



# Open Source Used In purchase 1.0.0- XX

## **Cisco Systems, Inc.**

[www.cisco.com](http://www.cisco.com)

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

Text Part Number: 78EE117C99-140385744

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

**In your requests please include the following reference number 78EE117C99-140385744**

## Contents

### **1.1 amqp-client 3.5.6**

1.1.1 Available under license

### **1.2 Apache Commons Lib Apache 2.0**

1.2.1 Available under license

### **1.3 Apache HttpClient 4.0.1**

1.3.1 Available under license

### **1.4 Apache HttpClient NIO 4.3.2**

1.4.1 Available under license

### **1.5 Apache Jakarta Commons HttpClient 3.1**

1.5.1 Available under license

### **1.6 Apache Log4j 1.2.16**

1.6.1 Available under license

### **1.7 Apache Log4j 1.2.17**

1.7.1 Available under license

### **1.8 apache-log4j 1.2.15**

1.8.1 Available under license

### **1.9 apache-log4j 1.2.15 :DUPLICATE**

1.9.1 Available under license

### **1.10 bcprov-jdk16 1.36**

1.10.1 Available under license

### **1.11 bsm-parser-api 3.56.0-1**

1.11.1 Available under license

### **1.12 common-mercury-data-objects 4.5.1-1**

1.12.1 Available under license

### **1.13 Commons IO 2.4**

1.13.1 Available under license

- 1.14 commons-codec 1.5 :commons-codec-1.5.jar**
  - 1.14.1 Available under license
- 1.15 commons-configuration 1.6**
  - 1.15.1 Available under license
- 1.16 commons-lang 2.6**
  - 1.16.1 Available under license
- 1.17 commons-lang3 3.3**
  - 1.17.1 Available under license
- 1.18 configuration-lib 1.0.3-1**
  - 1.18.1 Available under license
- 1.19 Crazybob - DebuggingObjectOutputStream 1.0**
  - 1.19.1 Available under license
- 1.20 distDES 0.1.0**
  - 1.20.1 Available under license
- 1.21 DOM 20001113**
  - 1.21.1 Available under license
- 1.22 Guava: Google Core Libraries for Java 18.0**
  - 1.22.1 Available under license
- 1.23 hornetq-commons 2.4.4.Final**
  - 1.23.1 Available under license
- 1.24 hornetq-core-client 2.2.14.Final**
  - 1.24.1 Available under license
- 1.25 hornetq-core-client 2.4.4.Final**
  - 1.25.1 Available under license
- 1.26 hornetq-jms-client 2.2.14.Final**
  - 1.26.1 Available under license
- 1.27 hornetq-jms-client 2.4.4.Final**
  - 1.27.1 Available under license
- 1.28 hornetq-journal 2.4.4.Final**
  - 1.28.1 Available under license
- 1.29 hornetq-native 2.4.4.Final**
  - 1.29.1 Available under license
- 1.30 http-server-jetty 1.1.1-2**
  - 1.30.1 Available under license
- 1.31 httpclient 4.2.3**
  - 1.31.1 Available under license
- 1.32 Jackson-annotations 2.2.2**
  - 1.32.1 Available under license
- 1.33 Jackson-core 2.2.2**

- 1.33.1 Available under license
- 1.34 jackson-databind 2.5.2**
  - 1.34.1 Available under license
- 1.35 jackson-databind 2.2.2**
  - 1.35.1 Available under license
- 1.36 Jackson-JAXRS-base 2.2.2**
- 1.37 Jackson-JAXRS-JSON 2.2.2**
- 1.38 Jackson-module-JAXB-annotations 2.2.2**
  - 1.38.1 Available under license
- 1.39 Java Servlet API 3.1.0**
  - 1.39.1 Available under license
- 1.40 javax.inject 1**
  - 1.40.1 Available under license
- 1.41 JAX-RS Core API 3.0.4.Final**
  - 1.41.1 Available under license
- 1.42 jboss-jms-api\_2.0\_spec 1.0.0.Final**
  - 1.42.1 Available under license
- 1.43 jetty-continuation 9.3.1.v20150714**
  - 1.43.1 Available under license
- 1.44 jetty-http 9.3.1.v20150714**
  - 1.44.1 Available under license
- 1.45 jetty-io 9.3.1.v20150714**
  - 1.45.1 Available under license
- 1.46 jetty-security 9.3.1.v20150714**
  - 1.46.1 Available under license
- 1.47 jetty-server 9.3.1.v20150714**
  - 1.47.1 Available under license
- 1.48 jetty-servlet 9.3.1.v20150714**
  - 1.48.1 Available under license
- 1.49 jetty-servlets 9.3.1.v20150714**
  - 1.49.1 Available under license
- 1.50 jetty-util 9.3.1.v20150714**
  - 1.50.1 Available under license
- 1.51 jgroups 3.3.4.Final**
  - 1.51.1 Available under license
- 1.52 JMS 1.1 1.1.1**
  - 1.52.1 Available under license
- 1.53 Joda convert 1.2**
  - 1.53.1 Available under license

## **1.54 Joda-Time 1.6.2**

1.54.1 Available under license

## **1.55 Joda-Time 2.3**

1.55.1 Available under license

## **1.56 jsr311-api 1.1.1**

1.56.1 Available under license

## **1.57 JUL to SLF4J bridge 1.7.7**

1.57.1 Available under license

## **1.58 jul-to-slf4j 1.7.7**

1.58.1 Available under license

## **1.59 jumi-actors 0.1.196**

1.59.1 Available under license

## **1.60 log4j 1.2.16**

1.60.1 Available under license

## **1.61 master-client-api 2.0.3-0**

1.61.1 Available under license

## **1.62 Metrics Core 3.0.2**

1.62.1 Available under license

## **1.63 netty 3.2.9.Final**

1.63.1 Available under license

## **1.64 netty-all 4.0.13.Final**

1.64.1 Available under license

## **1.65 nm-mercury-data-objects 4.2.0-1**

1.65.1 Available under license

## **1.66 oasm-mercury-data-objects 1.9.0-2**

1.66.1 Available under license

## **1.67 offercacher-application 3.58.0-2**

1.67.1 Available under license

## **1.68 quartz 1.7.3**

1.68.1 Available under license

## **1.69 scala-actors 2.10.2**

1.69.1 Available under license

## **1.70 scala-library 2.10.2**

1.70.1 Available under license

## **1.71 scala-reflect 2.10.2**

1.71.1 Available under license

## **1.72 Simple Logging Facade for Java (SLF4J) 1.7.2**

1.72.1 Available under license

## **1.73 SLF4J API Module 1.6.1**

- 1.73.1 Available under license
- 1.74 Spring Framework 2.0.7**
  - 1.74.1 Available under license
- 1.75 Spring JMS 4.0.0.RELEASE**
  - 1.75.1 Available under license
- 1.76 Spring TestContext Framework 4.0.0.RELEASE**
  - 1.76.1 Available under license
- 1.77 Spring Web 4.0.0.RELEASE**
  - 1.77.1 Available under license
- 1.78 spring-beans 3.2.1.RELEASE**
  - 1.78.1 Available under license
- 1.79 spring-core 3.1.1.RELEASE**
  - 1.79.1 Available under license
- 1.80 spring-core 4.0.0.Release**
  - 1.80.1 Available under license
- 1.81 spring-test 4.1.0**
  - 1.81.1 Available under license
- 1.82 spring-web 4.0.0.Release**
  - 1.82.1 Available under license
- 1.83 spring-web 4.2.3.RELEASE**
  - 1.83.1 Available under license
- 1.84 Xalan 2.7.1**
  - 1.84.1 Available under license

## 1.1 amqp-client 3.5.6

### 1.1.1 Available under license :

License names:

ASL 2.0

GPL v2

MPL 1.1

## 1.2 Apache Commons Lib Apache 2.0

### 1.2.1 Available under license :

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of



the Derivative Works; and

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

See the License for the specific language governing permissions and limitations under the License.

Apache Commons Lang

Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

Apache Commons Logging

Copyright 2003-2007 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

Apache Jakarta Commons Validator

Copyright 2001-2006 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

Apache Commons FileUpload

Copyright 2002-2008 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

Apache Commons Digester

Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

Apache Commons Collections

Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

Apache Commons Codec

Copyright 2002-2009 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

-----  
src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains  
test data from <http://aspell.sourceforge.net/test/batch0.tab>.

Copyright (C) 2002 Kevin Atkinson ([kevina@gnu.org](mailto:kevina@gnu.org)). Verbatim copying  
and distribution of this entire article is permitted in any medium,  
provided this notice is preserved.

-----  
Apache Commons BeanUtils

Copyright 2000-2010 The Apache Software Foundation

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

# 1.3 Apache HttpAsyncClient 4.0.1

## 1.3.1 Available under license :

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

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

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

"Derivative Works" shall mean any work, whether in Source or Object

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

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

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

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

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

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

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache HttpClient

Copyright 2010-2014 The Apache Software Foundation

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

## 1.4 Apache HttpClient NIO 4.3.2

### 1.4.1 Available under license :

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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



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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
  
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

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

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

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

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

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

9. Accepting Warranty or Additional Liability. While redistributing

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

#### END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache HttpCore NIO

Copyright 2005-2014 The Apache Software Foundation

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

## 1.5 Apache Jakarta Commons HttpClient 3.1

## 1.5.1 Available under license :

Apache Jakarta HttpClient  
Copyright 1999-2007 The Apache Software Foundation

This product includes software developed by  
The Apache Software Foundation (<http://www.apache.org/>).  
Apache Jakarta HttpClient  
Copyright 1999-2007 The Apache Software Foundation

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

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

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

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

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

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

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

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION



## 1. Definitions.

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

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

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

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

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

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
  
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## 1.6 Apache Log4j 1.2.16

### 1.6.1 Available under license :

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

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

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

## END OF TERMS AND CONDITIONS

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

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

Copyright 1999-2005 The Apache Software Foundation

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

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

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

Apache log4j

Copyright 2007 The Apache Software Foundation

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



# 1.7 Apache Log4j 1.2.17

## 1.7.1 Available under license :

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

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

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

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

separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

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

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

Copyright 1999-2005 The Apache Software Foundation

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

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

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

Apache log4j

Copyright 2007 The Apache Software Foundation

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

/\*

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

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

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

\*/

## 1.8 apache-log4j 1.2.15

### 1.8.1 Available under license :

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

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

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

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

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

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

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

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

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

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

8. Limitation of Liability. In no event and under no legal theory,



whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

#### END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache log4j  
Copyright 2007 The Apache Software Foundation

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

/\*

- \* Licensed to the Apache Software Foundation (ASF) under one or more
- \* contributor license agreements. See the NOTICE file distributed with
- \* this work for additional information regarding copyright ownership.
- \* The ASF licenses this file to You under the Apache License, Version 2.0
- \* (the "License"); you may not use this file except in compliance with
- \* the License. You may obtain a copy of the License at
- \*
- \* <http://www.apache.org/licenses/LICENSE-2.0>
- \*
- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS,
- \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- \* See the License for the specific language governing permissions and
- \* limitations under the License.
- \*/

## 1.9 apache-log4j 1.2.15 :DUPLICATE

### 1.9.1 Available under license :

Apache License

Version 2.0, January 2004

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
  
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

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

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

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

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

Copyright 1999-2005 The Apache Software Foundation

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

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

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

Apache log4j

Copyright 2007 The Apache Software Foundation

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

## 1.10 bcprov-jdk16 1.36

### 1.10.1 Available under license :

- \* The Bouncy Castle License

- \*

- \* Copyright (c) 2000-2006 The Legion Of The Bouncy Castle (<http://www.bouncycastle.org>)

```

* <p>
* Permission is hereby granted, free of charge, to any person obtaining a copy of this software
* and associated documentation files (the "Software"), to deal in the Software without restriction,
* including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
* and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
* subject to the following conditions:
* <p>
* The above copyright notice and this permission notice shall be included in all copies or substantial
* portions of the Software.
* <p>
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED,
* INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR
* PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
CONTRACT, TORT OR
* OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER
* DEALINGS IN THE SOFTWARE.
/* Partially optimised Serpent S Box boolean functions derived */
/* using a recursive descent analyser but without a full search */
/* of all subtrees. This set of S boxes is the result of work */
/* by Sam Simpson and Brian Gladman using the spare time on a */
/* cluster of high capacity servers to search for S boxes with */
/* this customised search engine. There are now an average of */
/* 15.375 terms per S box. */
/* */
/* Copyright: Dr B. R Gladman (gladman@seven77.demon.co.uk) */
/* and Sam Simpson (s.simpson@mia.co.uk) */
/* 17th December 1998 */
/* */
/* We hereby give permission for information in this file to be */
/* used freely subject only to acknowledgement of its origin. */

```

## 1.11 bsm-parser-api 3.56.0-1

### 1.11.1 Available under license :

NDS Proprietary  
 Copyright (c) 2010: NDS Limited  
 PROPRIETARY & CONFIDENTIAL

The copyright of this code and related documentation together with any other associated intellectual property rights are vested in NDS Limited and may not be used except in accordance with the terms of the licence that you have entered into with NDS Limited. Use of this material without an express licence from NDS Limited shall be an infringement of copyright and any other intellectual property rights that may be incorporated with this

material.

## 1.12 common-mercury-data-objects 4.5.1-1

### 1.12.1 Available under license :

NDS Proprietary

Copyright (c) 2010: NDS Limited

PROPRIETARY & CONFIDENTIAL

The copyright of this code and related documentation together with any other associated intellectual property rights are vested in NDS Limited and may not be used except in accordance with the terms of the licence that you have entered into with NDS Limited. Use of this material without an express licence from NDS Limited shall be an infringement of copyright and any other intellectual property rights that may be incorporated with this material.

## 1.13 Commons IO 2.4

### 1.13.1 Available under license :

Apache License

Version 2.0, January 2004

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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



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

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

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

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

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

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

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

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

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

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache Commons IO

Copyright 2002-2012 The Apache Software Foundation

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

/\*

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

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

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

\* limitations under the License.

\*/

# 1.14 commons-codec 1.5 :commons-codec-1.5.jar

## 1.14.1 Available under license :

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

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate



comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache Commons Codec

Copyright 2002-2011 The Apache Software Foundation

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

-----  
src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains  
test data from <http://aspell.sourceforge.net/test/batch0.tab>.

Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org). Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

-----  
/\*

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

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

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

\*/

# 1.15 commons-configuration 1.6

## 1.15.1 Available under license :

Apache License

Version 2.0, January 2004

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

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written

communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

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

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

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

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

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

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any

additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

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

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

END OF TERMS AND CONDITIONS

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

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "[ ]"

replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

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

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Apache Commons Configuration

Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by

The Apache Software Foundation (<http://www.apache.org/>).

## 1.16 commons-lang 2.6

### 1.16.1 Available under license :

Apache License

Version 2.0, January 2004

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
  
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

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

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer,



and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache Commons Lang

Copyright 2001-2011 The Apache Software Foundation

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

/\*

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

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

## 1.17 commons-lang3 3.3

### 1.17.1 Available under license :

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache Commons Lang

Copyright 2001-2014 The Apache Software Foundation

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

This product includes software from the Spring Framework,  
under the Apache License 2.0 (see: `StringUtils.containsWhitespace()`)

## 1.18 configuration-lib 1.0.3-1

### 1.18.1 Available under license :

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

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

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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained



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

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

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

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

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

# 1.19 Crazybob -

## DebuggingObjectOutputStream 1.0

### 1.19.1 Available under license :

Apache License

Version 2.0, January 2004

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

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

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

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

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

Licensors for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

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

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

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

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

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

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any

additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

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

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

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

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

Copyright [yyyy] [name of copyright owner]

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

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

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

## 1.20 distDES 0.1.0

### 1.20.1 Available under license :

/\*\*

- \* Copyright (c) 2005, Daniel Sparks, Xyclops Software Systems
- \* All rights reserved.
- \*
- \* Redistribution and use in source and binary forms, with or without
- \* modification, are permitted provided that the following conditions
- \* are met:
- \* Redistributions of source code must retain the above copyright notice,
- \* this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright
- \* notice, this list of conditions and the following disclaimer
- \* in the documentation and/or other materials provided with the
- \* distribution.
- \* Neither the name of Xyclops Software Systems nor the names of its
- \* contributors may be used to endorse or promote products derived from
- \* this software without specific prior written permission.
- \*
- \* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
- \* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
- \* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
- \* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
- \*
- \* COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
- \* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
- \* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
- \* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
- \* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
- \* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
- \* ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
- \* POSSIBILITY OF SUCH DAMAGE.
- \*/

## 1.21 DOM 20001113

## 1.21.1 Available under license :

/\*

- \* Copyright (c) 2000 World Wide Web Consortium,
- \* (Massachusetts Institute of Technology, Institut National de
- \* Recherche en Informatique et en Automatique, Keio University). All
- \* Rights Reserved. This program is distributed under the W3C's Software
- \* Intellectual Property License. This program is distributed in the
- \* hope that it will be useful, but WITHOUT ANY WARRANTY; without even
- \* the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
- \* PURPOSE.
- \* See W3C License <http://www.w3.org/Consortium/Legal/> for more details.

\*/

W3C Intellectual Rights Notice and Legal Disclaimers

This page includes notices and disclaimers related to:

Copyright

Notices

Trademarks

Disclaimers and Liabilities

See also Licenses for W3C Test Suites

Copyright 1994-2009 W3C (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University), All Rights Reserved.

World Wide Web Consortium (W3C) web site pages may contain other proprietary notices and copyright information, the terms of which must be observed and followed. Specific notices do exist for W3C documents and software. Also, there are specific usage policies associated with some of the W3C Icons. Please see our Intellectual Rights FAQ for common questions about using materials from our site.

Notice and Disclaimers

1. Unless otherwise noted, all materials contained in this Site are copyrighted and may not be used except as provided in these terms and conditions or in the copyright notice (documents and software) or other proprietary notice provided with the relevant materials.
2. The materials contained in the Site may be downloaded or copied provided that ALL copies retain the copyright and any other proprietary notices contained on the materials. No material may be modified, edited or taken out of context such that its use creates a false or misleading statement or impression as to the positions, statements or actions of W3C.
3. The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the Web site, its content, specifications, or software without specific, written prior permission. Title to copyright in Web site documents will at all times remain with copyright holders. Use of W3C trademarks and service marks is covered by the W3C Trademark and Servicemark License.
4. Caches of W3C materials should comply with the "maximum time to live" information provided with the materials. After such materials have expired they should not be served from caches without first validating the contents of the W3C Site. Organizations that want to mirror W3C content must abide by the W3C Mirroring Policy.

W3C Trademarks and Generic Terms

Trademarks owned by W3C host institutions on behalf of W3C and generic terms used by the W3C

5. The trademarks, logos, and service marks (collectively the "Trademarks") displayed on the Site are registered and unregistered Trademarks of the Massachusetts Institute of Technology (MIT), European Research Consortium for Informatics and Mathematics (ERCIM), or Keio University (Keio). All use of the W3C Trademarks is governed by the W3C Trademark and Servicemark License. No additional rights are granted by implication, estoppel, or otherwise. Terms which claimed as generic are not governed by any W3C license and are used as common descriptors by the W3C.

The following is a list of W3C terms claimed as a trademark or generic term by MIT, ERCIM, and/or Keio on behalf of the W3C:

W3C, World Wide Web Consortium (registered in numerous countries)

Amayaö, a Web Browser/Editor

ACSS (generic), Aural Cascading Style Sheets

CSS (generic), Cascading Style Sheets Specification

DOM (generic), Document Object Model

DSig (generic), Digital Signature Initiative

HTML (generic), HyperText Markup Language

HTTP (generic), Hypertext Transfer Protocol

JEP (generic)I, Joint Electronic Payment Initiative

Jigsawö, Java Web Server

MathML (generic), Mathematical Markup Language

Metadata (generic)

P3P, Platform for Privacy Preferences Project

PICS (generic), Platform for Internet Content Selection

PICSRules (generic), Rules Language for PICS

RDF (generic), Resource Description Framework

SMIL (generic), Synchronized Multimedia Integration Language

SVG (generic), Scalable Vector Graphics

WAIö, Web Accessibility Initiative

WebFonts (generic)

XENC (generic), XML Encryption

XHTML (generic), The Extensible HyperText Markup Language

XML (generic), Extensible Markup Language

XMLDSIG (generic), XML Signatures

XSL (generic), Extensible Stylesheet Language

The absence of a product or service name or logo from this list does not constitute a waiver of MIT's, ERCIM's, or Keio's trademark, ability to use a generic term, or other intellectual rights concerning that name or logo.

Any questions concerning the use, status, or standing of W3C trademarks should be directed to: [site-policy@w3.org](mailto:site-policy@w3.org) or to W3C, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139 USA.

Non-W3C Trademarks; Member Trademarks

The trademarks, logos, and service marks not owned on behalf of the W3C and that are displayed on the Site are the registered and unregistered marks of their respective owners. No rights are granted by the W3C to use such marks,



whether by implication, estoppel, or otherwise.

"METADATA" is a trademark of the Metadata Company. W3C uses the term "metadata" in a descriptive sense, meaning "data about data". W3C is not in any way affiliated with the Metadata Company.

#### Legal Disclaimers

6. W3C has not reviewed any or all of the web sites linked to this Site and is not responsible for the content of any off-site pages or any other web sites linked to this Site. Please understand that any non-W3C web site is independent from W3C, and W3C has no control over the content on that web site. In addition, a link to a non-W3C web site does not mean that W3C endorses or accepts any responsibility for the content, or the use, of such site. It is the user's responsibility to take precautions to ensure that whatever is selected is free of such items as viruses, worms, Trojan horses and other items of a destructive nature.

7. Information W3C publishes on its Site may contain references or cross references to W3C specifications, projects, programs and services that are not announced or available in your country. Such references do not imply that W3C intends to announce such specifications, projects, programs or services in your country.

8. Information on this Site may contain technical inaccuracies or typographical errors. Information may be changed or updated without notice. W3C may make improvements and/or changes in the materials contained in or described on this site at any time without notice. W3C may also make changes in these Terms and Conditions without notice. User is bound by such revisions and should therefore periodically visit this page to review the then current Terms and Conditions.

#### 9. Limitation on Warranties.

ALL MATERIALS ON THE W3C SITE ARE PROVIDED "AS IS." W3C, MIT, ERCIM, AND KEIO MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT. AS TO DOCUMENTS AND GRAPHICS PUBLISHED ON THIS SITE, W3C, MIT, ERCIM, AND KEIO MAKE NO REPRESENTATION OR WARRANTY THAT THE CONTENTS OF SUCH DOCUMENT OR GRAPHICS ARE FREE FROM ERROR OR SUITABLE FOR ANY PURPOSE; NOR THAT IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

Please note that some jurisdictions may not allow the exclusion of implied warranties, so some of the above exclusions may not apply to you.

#### 10. Limitation on Liability.

IN NO EVENT WILL W3C, MIT, ERCIM, AND KEIO BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES FOR ANY USE OF THIS SITE, OR ON ANY OTHER HYPERLINKED WEB SITE, INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS OR OTHER DATA ON YOUR INFORMATION HANDLING SYSTEM OR OTHERWISE, EVEN IF W3C, MIT, ERCIM, OR KEIO IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

---

Please see our Intellectual Rights FAQ for common questions about using materials from our site. Questions about this notice can be directed to [site-policy@w3.org](mailto:site-policy@w3.org).

Joseph Reagle <site-policy@w3.org>

Last revised \$Id: ipr-notice-20021231.html,v 1.18 2009/02/03 22:25:50 ijacobs Exp \$

Copyright 1998-2007 W3C (MIT,ERCIM, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

#### W3C SOFTWARE NOTICE AND LICENSE

<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>

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

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

The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.

Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

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

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

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

---

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

## 1.22 Guava: Google Core Libraries for Java

### 18.0

#### 1.22.1 Available under license :

```
/*
 * Copyright (C) 2009 The Guava Authors
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Apache License

Version 2.0, January 2004

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form,

including but not limited to compiled object code, generated documentation, and conversions to other media types.

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

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

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

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

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

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

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

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

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

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark,

and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

## 1.23 hornetq-commons 2.4.4.Final

### 1.23.1 Available under license :

ASL 2.0

<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.24 hornetq-core-client 2.2.14.Final

### 1.24.1 Available under license :

Apache 2

<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.25 hornetq-core-client 2.4.4.Final

### 1.25.1 Available under license :

ASL 2.0

<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.26 hornetq-jms-client 2.2.14.Final

### 1.26.1 Available under license :

Apache 2

<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.27 hornetq-jms-client 2.4.4.Final

### 1.27.1 Available under license :

ASL 2.0

<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.28 hornetq-journal 2.4.4.Final

### 1.28.1 Available under license :

ASL 2.0

<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.29 hornetq-native 2.4.4.Final

### 1.29.1 Available under license :

ASL 2.0

<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.30 http-server-jetty 1.1.1-2

### 1.30.1 Available under license :

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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



such litigation is filed.

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

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

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

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

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

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

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

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

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

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

such damages.

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

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

# 1.31 httpclient 4.2.3

## 1.31.1 Available under license :

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

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

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

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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

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

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

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

#### END OF TERMS AND CONDITIONS

Apache HttpComponents HttpClient  
Copyright 1999-2013 The Apache Software Foundation

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

/\*

\* =====

\*

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

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

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

\* =====

\*

\* This software consists of voluntary contributions made by many  
\* individuals on behalf of the Apache Software Foundation. For more  
\* information on the Apache Software Foundation, please see  
\* <http://www.apache.org/>.

\*  
\*/

## 1.32 Jackson-annotations 2.2.2

### 1.32.1 Available under license :

Apache License  
Version 2.0, January 2004

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written

communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

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

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

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

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

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

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for



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

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

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

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

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

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

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

Copyright [yyyy] [name of copyright owner]

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

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

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

This copy of Jackson JSON processor annotations is licensed under the  
Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the  
specific rights regarding derivate works.

You may obtain a copy of the License at:

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

## 1.33 Jackson-core 2.2.2

### 1.33.1 Available under license :

Apache License

Version 2.0, January 2004

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such

third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

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

third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

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

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

See the License for the specific language governing permissions and  
limitations under the License.

This copy of Jackson JSON processor streaming parser/generator is licensed under the  
Apache (Software) License, version 2.0 ("the License").

See the License for details about distribution rights, and the  
specific rights regarding derivate works.

You may obtain a copy of the License at:

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

# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.  
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has  
been in development since 2007.

It is currently developed by a community of developers, as well as supported  
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.  
To find the details that apply to this artifact see the accompanying LICENSE file.  
For more information, including possible other licensing options, contact  
FasterXML.com (<http://fasterxml.com>).

## Credits

A list of contributors may be found from CREDITS file, which is included  
in some artifacts (usually source distributions); but is always available  
from the source code management (SCM) system project uses.

## 1.34 jackson-databind 2.5.2

## 1.34.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License").  
See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>  
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.  
It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

### ## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (<http://fasterxml.com>).

### ## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

# 1.35 jackson-databind 2.2.2

## 1.35.1 Available under license :

Apache License

Version 2.0, January 2004

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

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

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

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

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

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

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

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

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

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

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

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



any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

This copy of Jackson JSON processor databind module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>  
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library. It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has been in development since 2007.

It is currently developed by a community of developers, as well as supported commercially by FasterXML.com.

## ## Licensing

Jackson core and extension components may be licensed under different licenses. To find the details that apply to this artifact see the accompanying LICENSE file. For more information, including possible other licensing options, contact FasterXML.com (<http://fasterxml.com>).

## ## Credits

A list of contributors may be found from CREDITS file, which is included in some artifacts (usually source distributions); but is always available from the source code management (SCM) system project uses.

/\* Jackson JSON-processor.

\*

\* Copyright (c) 2007- Tatu Saloranta, tatu.saloranta@iki.fi

\*

\* Licensed under the License specified in file LICENSE, included with the source code and binary code bundles.

\* You may not use this file except in compliance with the License.

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

## 1.36 Jackson-JAXRS-base 2.2.2

## 1.37 Jackson-JAXRS-JSON 2.2.2

## 1.38 Jackson-module-JAXB-annotations 2.2.2

### 1.38.1 Available under license :

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

See the License for the specific language governing permissions and limitations under the License.

## 1.39 Java Servlet API 3.1.0

### 1.39.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

#### 1. Definitions.

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

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

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

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

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

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

1.7. License. means this document.

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

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

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

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

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

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and

(b) associated documentation included in or with such code.

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

## 2. License Grants.

### 2.1. The Initial Developer Grant.

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

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

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

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

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

### 2.2. Contributor Grant.

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

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

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

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



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

### 3. Distribution Obligations.

#### 3.1. Availability of Source Code.

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

#### 3.2. Modifications.

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

#### 3.3. Required Notices.

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

#### 3.4. Application of Additional Terms.

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

#### 3.5. Distribution of Executable Versions.

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

#### 3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of

this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

#### 4. Versions of the License.

##### 4.1. New Versions.

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

##### 4.2. Effect of New Versions.

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

##### 4.3. Modified Versions.

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

#### 5. DISCLAIMER OF WARRANTY.

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

#### 6. TERMINATION.

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

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if

the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

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

## 7. LIMITATION OF LIABILITY.

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

## 8. U.S. GOVERNMENT END USERS.

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

## 9. MISCELLANEOUS.

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

make available any Covered Software.

## 10. RESPONSIBILITY FOR CLAIMS.

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

### NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

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

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

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

#### Preamble

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

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License.

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

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

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

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

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

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

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

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

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

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

number of this License, you may choose any version ever published by the Free Software Foundation.

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

#### NO WARRANTY

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

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

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

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

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

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

Copyright (C)

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



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

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

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

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

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

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

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

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

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

## "CLASSPATH" EXCEPTION TO THE GPL VERSION 2

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

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

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

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked

independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

/\*

\* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

\*

\* Copyright (c) 2008-2013 Oracle and/or its affiliates. All rights reserved.

\*

\* The contents of this file are subject to the terms of either the GNU

\* General Public License Version 2 only ("GPL") or the Common Development

\* and Distribution License("CDDL") (collectively, the "License"). You

\* may not use this file except in compliance with the License. You can

\* obtain a copy of the License at

\* [https://glassfish.dev.java.net/public/CDDL+GPL\\_1\\_1.html](https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html)

\* or packager/legal/LICENSE.txt. See the License for the specific

\* language governing permissions and limitations under the License.

\*

\* When distributing the software, include this License Header Notice in each

\* file and include the License file at packager/legal/LICENSE.txt.

\*

\* GPL Classpath Exception:

\* Oracle designates this particular file as subject to the "Classpath"

\* exception as provided by Oracle in the GPL Version 2 section of the License

\* file that accompanied this code.

\*

\* Modifications:

\* If applicable, add the following below the License Header, with the fields

\* enclosed by brackets [] replaced by your own identifying information:

\* "Portions Copyright [year] [name of copyright owner]"

\*

\* Contributor(s):

\* If you wish your version of this file to be governed by only the CDDL or

\* only the GPL Version 2, indicate your decision by adding "[Contributor]

\* elects to include this software in this distribution under the [CDDL or GPL

\* Version 2] license." If you don't indicate a single choice of license, a

\* recipient has the option to distribute your version of this file under

\* either the CDDL, the GPL Version 2 or to extend the choice of license to

\* its licensees as provided above. However, if you add GPL Version 2 code

\* and therefore, elected the GPL Version 2 license, then the option applies

\* only if the new code is made subject to such option by the copyright

\* holder.

\*/

## 1.40 javax.inject 1

## 1.40.1 Available under license :

Apache License

Version 2.0, January 2004

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

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product

names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

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

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

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

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

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

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

See the License for the specific language governing permissions and limitations under the License.

/\*  
\* Copyright (C) 2009 The JSR-330 Expert Group  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

## 1.41 JAX-RS Core API 3.0.4.Final

### 1.41.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

#### 1. Definitions.

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

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

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

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

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

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

1.7. "License" means this document.

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

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

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing

Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

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

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

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

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

## 2. License Grants.

### 2.1. The Initial Developer Grant.

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

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

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

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

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

### 2.2. Contributor Grant.

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

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce,

modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

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

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

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

### 3. Distribution Obligations.

#### 3.1. Availability of Source Code.

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

#### 3.2. Modifications.

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

#### 3.3. Required Notices.

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

#### 3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However,



you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

### 3.5. Distribution of Executable Versions.

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

### 3.6. Larger Works.

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

## 4. Versions of the License.

### 4.1. New Versions.

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

### 4.2. Effect of New Versions.

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

### 4.3. Modified Versions.

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

## 5. DISCLAIMER OF WARRANTY.

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

## 6. TERMINATION.

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

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

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

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

## 7. LIMITATION OF LIABILITY.

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

JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

#### 8. U.S. GOVERNMENT END USERS.

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

#### 9. MISCELLANEOUS.

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

#### 10. RESPONSIBILITY FOR CLAIMS.

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

#### NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

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

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

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

## Preamble

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

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

#### NO WARRANTY

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

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY

TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

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

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

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

Copyright (C) <year> <name of author>

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

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

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

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

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

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

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

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



signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

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

## "CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

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

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

/\*

\* DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

\*

\* Copyright (c) 2010-2013 Oracle and/or its affiliates. All rights reserved.

\*

\* The contents of this file are subject to the terms of either the GNU  
\* General Public License Version 2 only ("GPL") or the Common Development

\* and Distribution License("CDDL") (collectively, the "License"). You

\* may not use this file except in compliance with the License. You can

\* obtain a copy of the License at

\* [http://glassfish.java.net/public/CDDL+GPL\\_1\\_1.html](http://glassfish.java.net/public/CDDL+GPL_1_1.html)

\* or packager/legal/LICENSE.txt. See the License for the specific

\* language governing permissions and limitations under the License.

\*

\* When distributing the software, include this License Header Notice in each

\* file and include the License file at packager/legal/LICENSE.txt.

\*

\* GPL Classpath Exception:

\* Oracle designates this particular file as subject to the "Classpath"

\* exception as provided by Oracle in the GPL Version 2 section of the License

\* file that accompanied this code.

\*

\* Modifications:

\* If applicable, add the following below the License Header, with the fields

- \* enclosed by brackets [] replaced by your own identifying information:
- \* "Portions Copyright [year] [name of copyright owner]"
- \*
- \* Contributor(s):
- \* If you wish your version of this file to be governed by only the CDDL or
- \* only the GPL Version 2, indicate your decision by adding "[Contributor]
- \* elects to include this software in this distribution under the [CDDL or GPL
- \* Version 2] license." If you don't indicate a single choice of license, a
- \* recipient has the option to distribute your version of this file under
- \* either the CDDL, the GPL Version 2 or to extend the choice of license to
- \* its licensees as provided above. However, if you add GPL Version 2 code
- \* and therefore, elected the GPL Version 2 license, then the option applies
- \* only if the new code is made subject to such option by the copyright
- \* holder.
- \*/

## 1.42 jboss-jms-api\_2.0\_spec 1.0.0.Final

### 1.42.1 Available under license :

License names:

Common Development and Distribution License

GNU General Public License, Version 2 with the Classpath Exception

## 1.43 jetty-continuation 9.3.1.v20150714

### 1.43.1 Available under license :

License names:

Apache Software License - Version 2.0

Eclipse Public License - Version 1.0

## 1.44 jetty-http 9.3.1.v20150714

### 1.44.1 Available under license :

License names:

Apache Software License - Version 2.0

Eclipse Public License - Version 1.0

## 1.45 jetty-io 9.3.1.v20150714

### 1.45.1 Available under license :

License names:

Apache Software License - Version 2.0

Eclipse Public License - Version 1.0

## 1.46 jetty-security 9.3.1.v20150714

### 1.46.1 Available under license :

License names:

Apache Software License - Version 2.0

Eclipse Public License - Version 1.0

## 1.47 jetty-server 9.3.1.v20150714

### 1.47.1 Available under license :

License names:

Apache Software License - Version 2.0

Eclipse Public License - Version 1.0

## 1.48 jetty-servlet 9.3.1.v20150714

### 1.48.1 Available under license :

License names:

Apache Software License - Version 2.0

Eclipse Public License - Version 1.0

## 1.49 jetty-servlets 9.3.1.v20150714

### 1.49.1 Available under license :

License names:

Apache Software License - Version 2.0

Eclipse Public License - Version 1.0

## 1.50 jetty-util 9.3.1.v20150714

### 1.50.1 Available under license :

License names:

Apache Software License - Version 2.0

Eclipse Public License - Version 1.0

## 1.51 jgroups 3.3.4.Final

### 1.51.1 Available under license :

GNU Lesser General Public License 2.1

<http://www.opensource.org/licenses/lgpl-2.1.php>

# 1.52 JMS 1.1 1.1.1

## 1.52.1 Available under license :

Apache Geronimo  
Copyright 2003-2008 The Apache Software Foundation

This product includes software developed by  
The Apache Software Foundation (<http://www.apache.org/>).  
Apache License

Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

as of the date such litigation is filed.

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

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

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

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

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

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

## 1.53 Joda convert 1.2

### 1.53.1 Available under license :

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

"You" (or "Your") shall mean an individual or Legal Entity



exercising permissions granted by this License.

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

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

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

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

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

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

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

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

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

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

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

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

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

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

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

=====

= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

=====

This product includes software developed by  
Joda.org (<http://www.joda.org/>).

## 1.54 Joda-Time 1.6.2

### 1.54.1 Available under license :

Apache License

Version 2.0, January 2004

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

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

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

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

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

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

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

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

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

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

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works

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

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

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

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

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

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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



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

=====

= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

=====

This product includes software developed by Joda.org (<http://www.joda.org/>).

## 1.55 Joda-Time 2.3

### 1.55.1 Available under license :

Apache License

Version 2.0, January 2004

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

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

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

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

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

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

=====

= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =

=====

This product includes software developed by  
Joda.org (<http://www.joda.org/>).

## 1.56 jsr311-api 1.1.1

### 1.56.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0  
Thu, 2007-06-07 16:10 ? nelson

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

#### 1. Definitions.

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

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

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

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

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

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

1.7. "License" means this document.

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

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer

software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## 2. License Grants.

### 2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

## 2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

## 3. Distribution Obligations.

### 3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make



available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

### 3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

### 3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

### 3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

### 3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered

Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

### 3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

## 4. Versions of the License.

### 4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

### 4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software

available under the terms of any subsequent version of the License published by the license steward.

#### 4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

#### 5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

#### 6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant)

and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

## 7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

## 8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

## 9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

## 10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

\* Login or register to post comments

# 1.57 JUL to SLF4J bridge 1.7.7

## 1.57.1 Available under license :

```
cat jul-to-slf4j-1.7.7-sources/org/slf4j/bridge/SLF4JBridgeHandler.java
```

```
/**
```

```
* Copyright (c) 2004-2011 QOS.ch
```

```
* All rights reserved.
```

```
*
```

```
* Permission is hereby granted, free of charge, to any person obtaining
```

```
* a copy of this software and associated documentation files (the
```

```
* "Software"), to deal in the Software without restriction, including
```

\* without limitation the rights to use, copy, modify, merge, publish,  
\* distribute, sublicense, and/or sell copies of the Software, and to  
\* permit persons to whom the Software is furnished to do so, subject to  
\* the following conditions:  
\*  
\* The above copyright notice and this permission notice shall be  
\* included in all copies or substantial portions of the Software.  
\*  
\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
\* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
\* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
\* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE  
\* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION  
\* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
\* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
\*  
\*/

## 1.58 jul-to-slf4j 1.7.7

### 1.58.1 Available under license :

```
/**  
 * Copyright (c) 2004-2011 QOS.ch  
 * All rights reserved.  
 *  
 * Permission is hereby granted, free of charge, to any person obtaining  
 * a copy of this software and associated documentation files (the  
 * "Software"), to deal in the Software without restriction, including  
 * without limitation the rights to use, copy, modify, merge, publish,  
 * distribute, sublicense, and/or sell copies of the Software, and to  
 * permit persons to whom the Software is furnished to do so, subject to  
 * the following conditions:  
 *  
 * The above copyright notice and this permission notice shall be  
 * included in all copies or substantial portions of the Software.  
 *  
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
 * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE  
 * LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION  
 * OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
 * WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
 *  
 */
```

# 1.59 jumi-actors 0.1.196

## 1.59.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensors for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of



any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

#### APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

// Copyright © 2011-2012, Esko Luontola <[www.orfjackal.net](http://www.orfjackal.net)>

// This software is released under the Apache License 2.0.

// The license text is at <http://www.apache.org/licenses/LICENSE-2.0>

## 1.60 log4j 1.2.16

### 1.60.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]"

replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache log4j

Copyright 2007 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

## 1.61 master-client-api 2.0.3-0

### 1.61.1 Available under license :

NDS Proprietary

Copyright (c) 2010: NDS Limited

PROPRIETARY & CONFIDENTIAL

The copyright of this code and related documentation together with any other associated intellectual property rights are vested in NDS Limited and may not be used except in accordance with the terms of the licence that you have entered into with NDS Limited. Use of this material without an express licence from NDS Limited shall be an infringement of copyright and any other intellectual property rights that may be incorporated with this material.

## 1.62 Metrics Core 3.0.2

### 1.62.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works



that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2010-2012 Coda Hale and Yammer, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Metrics

Copyright 2010-2013 Coda Hale and Yammer, Inc.

This product includes software developed by Coda Hale and Yammer, Inc.

This product includes code derived from the JSR-166 project (ThreadLocalRandom, Striped64, LongAdder), which was released with the following comments:

Written by Doug Lea with assistance from members of JCP JSR-166 Expert Group and released to the public domain, as explained at <http://creativecommons.org/publicdomain/zero/1.0/>

## 1.63 netty 3.2.9.Final

### 1.63.1 Available under license :

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

The Netty Project  
=====

Please visit the Netty web site for more information:

\* <http://www.jboss.org/netty/>

Copyright 2009 Red Hat, Inc.

Red Hat licenses this product to you under the Apache License, version 2.0 (the "License"); you may not use this product except in compliance with the License. You may obtain a copy of the License at:

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed

under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in the 'license' directory of the distribution file, for the license terms of the components that this product depends on.

-----  
This product contains the extensions to Java Collections Framework which has been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

\* LICENSE:

\* license/LICENSE.jsr166y.txt (Public Domain)

\* HOMEPAGE:

\* <http://gee.cs.oswego.edu/cgi-bin/viewcvcs.cgi/jsr166/>

\* <http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jboss/cache/experimental/jsr166/>

This product contains a modified version of Robert Harder's Public Domain Base64 Encoder and Decoder, which can be obtained at:

\* LICENSE:

\* license/LICENSE.base64.txt (Public Domain)

\* HOMEPAGE:

\* <http://iharder.sourceforge.net/current/java/base64/>

This product contains a modified version of 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

\* LICENSE:

\* license/LICENSE.jzlib.txt (BSD Style License)

\* HOMEPAGE:

\* <http://www.jcraft.com/jzlib/>

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

\* LICENSE:

\* license/LICENSE.protobuf.txt (New BSD License)

\* HOMEPAGE:

\* <http://code.google.com/p/protobuf/>

This product optionally depends on 'SLF4J', a simple logging facade for Java, which can be obtained at:

\* LICENSE:

\* license/LICENSE.slf4j.txt (MIT License)

\* HOMEPAGE:



\* <http://www.slf4j.org/>

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

\* LICENSE:

\* [license/LICENSE.commons-logging.txt](#) (Apache License 2.0)

\* HOMEPAGE:

\* <http://commons.apache.org/logging/>

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

\* LICENSE:

\* [license/LICENSE.log4j.txt](#) (Apache License 2.0)

\* HOMEPAGE:

\* <http://logging.apache.org/log4j/>

This product optionally depends on 'JBoss Logging', a logging framework, which can be obtained at:

\* LICENSE:

\* [license/LICENSE.jboss-logging.txt](#) (GNU LGPL 2.1)

\* HOMEPAGE:

\* <http://anonsvn.jboss.org/repos/common/common-logging-spi/>

This product optionally depends on 'Apache Felix', an open source OSGi framework implementation, which can be obtained at:

\* LICENSE:

\* [license/LICENSE.felix.txt](#) (Apache License 2.0)

\* HOMEPAGE:

\* <http://felix.apache.org/>

The Netty Project - Copyright 2009 Red Hat, Inc, and is licensed under the Apache License version 2.0 as published by the Apache Software Foundation.

A summary of the individual contributors is given below. Any omission should be sent to Trustin Lee <[trustin@gmail.com](mailto:trustin@gmail.com)>.

SVN Login(s)	Name
amit.bhayani@jboss.com	Amit Bhayani
ataylor	Andy Taylor
beve	Daniel Bevenius
fredbregier	Frederic Bregier
trustin	Trustin Heuseung Lee

\* JBoss is a registered trademark of Red Hat, Inc.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicator holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
  
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

#### END OF TERMS AND CONDITIONS

Copyright 2008, Google Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- \* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever

copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuate of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

/\*

\* Copyright (c) 2004-2007 QOS.ch

\* All rights reserved.

\*

\* Permission is hereby granted, free of charge, to any person obtaining

\* a copy of this software and associated documentation files (the

\* "Software"), to deal in the Software without restriction, including

\* without limitation the rights to use, copy, modify, merge, publish,

\* distribute, sublicense, and/or sell copies of the Software, and to

\* permit persons to whom the Software is furnished to do so, subject to

\* the following conditions:

\*

\* The above copyright notice and this permission notice shall be

\* included in all copies or substantial portions of the Software.

\*

\* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,

\* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

\* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND

\* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

\* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION

\* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION

\* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*/

Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your  
freedom to share and change it. By contrast, the GNU General Public  
Licenses are intended to guarantee your freedom to share and change  
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some  
specially designated software packages--typically libraries--of the  
Free Software Foundation and other authors who decide to use it. You  
can use it too, but we suggest you first think carefully about whether  
this license or the ordinary General Public License is the better  
strategy to use in any particular case, based on the explanations below.



When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using

a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

#### GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License").

Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices

stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in

these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the

Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus



excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, 5th Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990  
Ty Coon, President of Vice

That's all there is to it!

# 1.64 netty-all 4.0.13.Final

## 1.64.1 Available under license :

```
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

```
/*
 * Written by Doug Lea with assistance from members of JCP JSR-166
 * Expert Group and released to the public domain, as explained at
 * http://creativecommons.org/publicdomain/zero/1.0/
 */
```

```
package io.netty.util.internal.chmv8;
```

```
import java.util.concurrent.RecursiveAction;
```

```
/**
 * A {@link ForkJoinTask} with a completion action performed when
 * triggered and there are no remaining pending actions.
 * CountedCompleters are in general more robust in the
 * presence of subtask stalls and blockage than are other forms of
 * ForkJoinTasks, but are less intuitive to program. Uses of
 * CountedCompleter are similar to those of other completion based
 * components (such as {@link java.nio.channels.CompletionHandler})
 * except that multiple <em>pending</em> completions may be necessary
 * to trigger the completion action {@link #onCompletion(CountedCompleter)},
 * not just one.
 * Unless initialized otherwise, the {@linkplain #getPendingCount pending
 * count} starts at zero, but may be (atomically) changed using
 * methods {@link #setPendingCount}, {@link #addToPendingCount}, and
 * {@link #compareAndSetPendingCount}. Upon invocation of {@link
 * #tryComplete}, if the pending action count is nonzero, it is
 * decremented; otherwise, the completion action is performed, and if
 * this completer itself has a completer, the process is continued
```

\* with its completer. As is the case with related synchronization  
\* components such as { @link java.util.concurrent.Phaser Phaser } and  
\* { @link java.util.concurrent.Semaphore Semaphore }, these methods  
\* affect only internal counts; they do not establish any further  
\* internal bookkeeping. In particular, the identities of pending  
\* tasks are not maintained. As illustrated below, you can create  
\* subclasses that do record some or all pending tasks or their  
\* results when needed. As illustrated below, utility methods  
\* supporting customization of completion traversals are also  
\* provided. However, because CountedCompleters provide only basic  
\* synchronization mechanisms, it may be useful to create further  
\* abstract subclasses that maintain linkages, fields, and additional  
\* support methods appropriate for a set of related usages.

\*  
\* <p>A concrete CountedCompleter class must define method { @link  
\* #compute }, that should in most cases (as illustrated below), invoke  
\* { @code tryComplete() } once before returning. The class may also  
\* optionally override method { @link #onCompletion(CountedCompleter) }  
\* to perform an action upon normal completion, and method  
\* { @link #onExceptionalCompletion(Throwable, CountedCompleter) } to  
\* perform an action upon any exception.

\*  
\* <p>CountedCompleters most often do not bear results, in which case  
\* they are normally declared as { @code CountedCompleter<Void> }, and  
\* will always return { @code null } as a result value. In other cases,  
\* you should override method { @link #getRawResult } to provide a  
\* result from { @code join(), invoke() }, and related methods. In  
\* general, this method should return the value of a field (or a  
\* function of one or more fields) of the CountedCompleter object that  
\* holds the result upon completion. Method { @link #setRawResult } by  
\* default plays no role in CountedCompleters. It is possible, but  
\* rarely applicable, to override this method to maintain other  
\* objects or fields holding result data.

\*  
\* <p>A CountedCompleter that does not itself have a completer (i.e.,  
\* one for which { @link #getCompleter } returns { @code null }) can be  
\* used as a regular ForkJoinTask with this added functionality.  
\* However, any completer that in turn has another completer serves  
\* only as an internal helper for other computations, so its own task  
\* status (as reported in methods such as { @link ForkJoinTask#isDone })  
\* is arbitrary; this status changes only upon explicit invocations of  
\* { @link #complete }, { @link ForkJoinTask#cancel },  
\* { @link ForkJoinTask#completeExceptionally(Throwable) } or upon  
\* exceptional completion of method { @code compute }. Upon any  
\* exceptional completion, the exception may be relayed to a task's  
\* completer (and its completer, and so on), if one exists and it has  
\* not otherwise already completed. Similarly, cancelling an internal  
\* CountedCompleter has only a local effect on that completer, so is

\* not often useful.

\*

\* **Sample Usages.**

\*

\* **Parallel recursive decomposition.** CountedCompleters may be arranged in trees similar to those often used with { [RecursiveAction](#) }s, although the constructions involved in setting them up typically vary. Here, the completer of each task is its parent in the computation tree. Even though they entail a bit more bookkeeping, CountedCompleters may be better choices when applying a possibly time-consuming operation (that cannot be further subdivided) to each element of an array or collection; especially when the operation takes a significantly different amount of time to complete for some elements than others, either because of intrinsic variation (for example I/O) or auxiliary effects such as garbage collection. Because CountedCompleters provide their own continuations, other threads need not block waiting to perform them.

\*

\* For example, here is an initial version of a class that uses divide-by-two recursive decomposition to divide work into single pieces (leaf tasks). Even when work is split into individual calls, tree-based techniques are usually preferable to directly forking leaf tasks, because they reduce inter-thread communication and improve load balancing. In the recursive case, the second of each pair of subtasks to finish triggers completion of its parent (because no result combination is performed, the default no-op implementation of method { `onCompletion` } is not overridden). A static utility method sets up the base task and invokes it (here, implicitly using the { [ForkJoinPool#commonPool\(\)](#) }).

\*

\* 

```
 { 
```

```
class MyOperation<E> { void apply(E e) { ... } }
```

\*

```
class ForEach<E> extends CountedCompleter<Void> {
```

\*

```
    public static <E> void forEach(E[] array, MyOperation<E> op) {
```

```
        new ForEach<E>(null, array, op, 0, array.length).invoke();
```

```
    }
```

\*

```
    final E[] array; final MyOperation<E> op; final int lo, hi;
```

```
    ForEach(CountedCompleter<?> p, E[] array, MyOperation<E> op, int lo, int hi) {
```

```
        super(p);
```

```
        this.array = array; this.op = op; this.lo = lo; this.hi = hi;
```

```
    }
```

\*

```
    public void compute() { // version 1
```

```
        if (hi - lo >= 2) {
```

```

*   int mid = (lo + hi) >>> 1;
*   setPendingCount(2); // must set pending count before fork
*   new ForEach(this, array, op, mid, hi).fork(); // right child
*   new ForEach(this, array, op, lo, mid).fork(); // left child
*   }
*   else if (hi > lo)
*       op.apply(array[lo]);
*   tryComplete();
*   }
* }}</pre>

```

\* This design can be improved by noticing that in the recursive case, the task has nothing to do after forking its right task, so can directly invoke its left task before returning. (This is an analog of tail recursion removal.) Also, because the task returns upon executing its left task (rather than falling through to invoke {`tryComplete`}) the pending count is set to one:

```

* <pre> { @code
* class ForEach<E> ...
*   public void compute() { // version 2
*     if (hi - lo >= 2) {
*       int mid = (lo + hi) >>> 1;
*       setPendingCount(1); // only one pending
*       new ForEach(this, array, op, mid, hi).fork(); // right child
*       new ForEach(this, array, op, lo, mid).compute(); // direct invoke
*     }
*     else {
*       if (hi > lo)
*         op.apply(array[lo]);
*       tryComplete();
*     }
*   }
* }
* }</pre>

```

\* As a further improvement, notice that the left task need not even exist. Instead of creating a new one, we can iterate using the original task, and add a pending count for each fork. Additionally, because no task in this tree implements an {[link #onCompletion\(CountedCompleter\)](#)} method, {`tryComplete()`} can be replaced with {[link #propagateCompletion](#)}.

```

* <pre> { @code
* class ForEach<E> ...
*   public void compute() { // version 3
*     int l = lo, h = hi;
*     while (h - l >= 2) {
*       int mid = (l + h) >>> 1;
*       addToPendingCount(1);

```

```

*   new ForEach(this, array, op, mid, h).fork(); // right child
*   h = mid;
*   }
*   if (h > 1)
*       op.apply(array[l]);
*   propagateCompletion();
*   }
* }</pre>
*
* Additional improvements of such classes might entail precomputing
* pending counts so that they can be established in constructors,
* specializing classes for leaf steps, subdividing by say, four,
* instead of two per iteration, and using an adaptive threshold
* instead of always subdividing down to single elements.
*
* <p><b>Searching.</b> A tree of CountedCompleters can search for a
* value or property in different parts of a data structure, and
* report a result in an { @link
* java.util.concurrent.atomic.AtomicReference AtomicReference } as
* soon as one is found. The others can poll the result to avoid
* unnecessary work. (You could additionally { @linkplain #cancel
* cancel } other tasks, but it is usually simpler and more efficient
* to just let them notice that the result is set and if so skip
* further processing.) Illustrating again with an array using full
* partitioning (again, in practice, leaf tasks will almost always
* process more than one element):
*
* <pre> { @code
* class Searcher<E> extends CountedCompleter<E> {
*   final E[] array; final AtomicReference<E> result; final int lo, hi;
*   Searcher(CountedCompleter<?> p, E[] array, AtomicReference<E> result, int lo, int hi) {
*     super(p);
*     this.array = array; this.result = result; this.lo = lo; this.hi = hi;
*   }
*   public E getRawResult() { return result.get(); }
*   public void compute() { // similar to ForEach version 3
*     int l = lo, h = hi;
*     while (result.get() == null && h >= l) {
*       if (h - l >= 2) {
*         int mid = (l + h) >>> 1;
*         addToPendingCount(1);
*         new Searcher(this, array, result, mid, h).fork();
*         h = mid;
*       }
*     }
*     else {
*       E x = array[l];
*       if (matches(x) && result.compareAndSet(null, x))
*         quietlyCompleteRoot(); // root task is now joinable

```

```

*     break;
*   }
* }
* tryComplete(); // normally complete whether or not found
* }
* boolean matches(E e) { ... } // return true if found
*
* public static <E> E search(E[] array) {
*   return new Searcher<E>(null, array, new AtomicReference<E>(), 0, array.length).invoke();
* }
* }}</pre>

```

\* In this example, as well as others in which tasks have no other effects except to compareAndSet a common result, the trailing unconditional invocation of { @code tryComplete } could be made conditional ( { @code if (result.get() == null) tryComplete(); } ) because no further bookkeeping is required to manage completions once the root task completes.

\* <p><b>Recording subtasks.</b> CountedCompleter tasks that combine results of multiple subtasks usually need to access these results in method { @link #onCompletion(CountedCompleter) }. As illustrated in the following class (that performs a simplified form of map-reduce where mappings and reductions are all of type { @code E } ), one way to do this in divide and conquer designs is to have each subtask record its sibling, so that it can be accessed in method { @code onCompletion }. This technique applies to reductions in which the order of combining left and right results does not matter; ordered reductions require explicit left/right designations. Variants of other streamlinings seen in the above examples may also apply.

```

* <pre> { @code
* class MyMapper<E> { E apply(E v) { ... } }
* class MyReducer<E> { E apply(E x, E y) { ... } }
* class MapReducer<E> extends CountedCompleter<E> {
*   final E[] array; final MyMapper<E> mapper;
*   final MyReducer<E> reducer; final int lo, hi;
*   MapReducer<E> sibling;
*   E result;
*   MapReducer(CountedCompleter<?> p, E[] array, MyMapper<E> mapper,
*     MyReducer<E> reducer, int lo, int hi) {
*     super(p);
*     this.array = array; this.mapper = mapper;
*     this.reducer = reducer; this.lo = lo; this.hi = hi;
*   }
*   public void compute() {
*     if (hi - lo >= 2) {
*       int mid = (lo + hi) >>> 1;

```



```

*   MapReducer<E> left = new MapReducer(this, array, mapper, reducer, lo, mid);
*   MapReducer<E> right = new MapReducer(this, array, mapper, reducer, mid, hi);
*   left.sibling = right;
*   right.sibling = left;
*   setPendingCount(1); // only right is pending
*   right.fork();
*   left.compute(); // directly execute left
* }
* else {
*   if (hi > lo)
*     result = mapper.apply(array[lo]);
*   tryComplete();
* }
* }
* public void onCompletion(CountedCompleter<?> caller) {
*   if (caller != this) {
*     MapReducer<E> child = (MapReducer<E>)caller;
*     MapReducer<E> sib = child.sibling;
*     if (sib == null || sib.result == null)
*       result = child.result;
*     else
*       result = reducer.apply(child.result, sib.result);
*   }
* }
* public E getRawResult() { return result; }
*
* public static <E> E mapReduce(E[] array, MyMapper<E> mapper, MyReducer<E> reducer) {
*   return new MapReducer<E>(null, array, mapper, reducer,
*     0, array.length).invoke();
* }
* }}</pre>

```

\* Here, method { @code onCompletion } takes a form common to many
 \* completion designs that combine results. This callback-style method
 \* is triggered once per task, in either of the two different contexts
 \* in which the pending count is, or becomes, zero: (1) by a task
 \* itself, if its pending count is zero upon invocation of { @code
 \* tryComplete }, or (2) by any of its subtasks when they complete and
 \* decrement the pending count to zero. The { @code caller } argument
 \* distinguishes cases. Most often, when the caller is { @code this },
 \* no action is necessary. Otherwise the caller argument can be used
 \* (usually via a cast) to supply a value (and/or links to other
 \* values) to be combined. Assuming proper use of pending counts, the
 \* actions inside { @code onCompletion } occur (once) upon completion of
 \* a task and its subtasks. No additional synchronization is required
 \* within this method to ensure thread safety of accesses to fields of
 \* this task or other completed tasks.

\* **Completion Traversals**. If using `onCompletion` to process completions is inapplicable or inconvenient, you can use methods `firstComplete` and `nextComplete` to create custom traversals. For example, to define a `MapReducer` that only splits out right-hand tasks in the form of the third `ForEach` example, the completions must cooperatively reduce along unexhausted subtask links, which can be done as follows:

```

* <pre> { @code
* class MapReducer<E> extends CountedCompleter<E> { // version 2
*   final E[] array; final MyMapper<E> mapper;
*   final MyReducer<E> reducer; final int lo, hi;
*   MapReducer<E> forks, next; // record subtask forks in list
*   E result;
*   MapReducer(CountedCompleter<?> p, E[] array, MyMapper<E> mapper,
*       MyReducer<E> reducer, int lo, int hi, MapReducer<E> next) {
*     super(p);
*     this.array = array; this.mapper = mapper;
*     this.reducer = reducer; this.lo = lo; this.hi = hi;
*     this.next = next;
*   }
*   public void compute() {
*     int l = lo, h = hi;
*     while (h - l >= 2) {
*       int mid = (l + h) >>> 1;
*       addToPendingCount(1);
*       (forks = new MapReducer(this, array, mapper, reducer, mid, h, forks)).fork();
*       h = mid;
*     }
*     if (h > l)
*       result = mapper.apply(array[l]);
*     // process completions by reducing along and advancing subtask links
*     for (CountedCompleter<?> c = firstComplete(); c != null; c = c.nextComplete()) {
*       for (MapReducer t = (MapReducer)c, s = t.forks; s != null; s = t.forks = s.next)
*         t.result = reducer.apply(t.result, s.result);
*     }
*   }
*   public E getRawResult() { return result; }
* }
*
* public static <E> E mapReduce(E[] array, MyMapper<E> mapper, MyReducer<E> reducer) {
*   return new MapReducer<E>(null, array, mapper, reducer,
*       0, array.length, null).invoke();
* }
* }</pre>

```

\* **Triggers**. Some `CountedCompleters` are themselves never forked, but instead serve as bits of plumbing in other designs; including those in which the completion of one or more async tasks

```

* triggers another async task. For example:
*
* <pre> { @code
* class HeaderBuilder extends CountedCompleter<...> { ... }
* class BodyBuilder extends CountedCompleter<...> { ... }
* class PacketSender extends CountedCompleter<...> {
*   PacketSender(...) { super(null, 1); ... } // trigger on second completion
*   public void compute() { } // never called
*   public void onCompletion(CountedCompleter<?> caller) { sendPacket(); }
* }
* // sample use:
* PacketSender p = new PacketSender();
* new HeaderBuilder(p, ...).fork();
* new BodyBuilder(p, ...).fork();
* }</pre>
*
* @since 1.8
* @author Doug Lea
*/
@SuppressWarnings("all")
public abstract class CountedCompleter<T> extends ForkJoinTask<T> {
    private static final long serialVersionUID = 5232453752276485070L;

    /** This task's completer, or null if none */
    final CountedCompleter<?> completer;
    /** The number of pending tasks until completion */
    volatile int pending;

    /**
     * Creates a new CountedCompleter with the given completer
     * and initial pending count.
     *
     * @param completer this task's completer, or {@code null} if none
     * @param initialPendingCount the initial pending count
     */
    protected CountedCompleter(CountedCompleter<?> completer,
                               int initialPendingCount) {
        this.completer = completer;
        this.pending = initialPendingCount;
    }

    /**
     * Creates a new CountedCompleter with the given completer
     * and an initial pending count of zero.
     *
     * @param completer this task's completer, or {@code null} if none
     */
    protected CountedCompleter(CountedCompleter<?> completer) {

```

```

    this.completer = completer;
}

/**
 * Creates a new CountedCompleter with no completer
 * and an initial pending count of zero.
 */
protected CountedCompleter() {
    this.completer = null;
}

/**
 * The main computation performed by this task.
 */
public abstract void compute();

/**
 * Performs an action when method {@link #tryComplete} is invoked
 * and the pending count is zero, or when the unconditional
 * method {@link #complete} is invoked. By default, this method
 * does nothing. You can distinguish cases by checking the
 * identity of the given caller argument. If not equal to {@code
 * this}, then it is typically a subtask that may contain results
 * (and/or links to other results) to combine.
 *
 * @param caller the task invoking this method (which may
 * be this task itself)
 */
public void onCompletion(CountedCompleter<?> caller) {
}

/**
 * Performs an action when method {@link
 * #completeExceptionally(Throwable)} is invoked or method {@link
 * #compute} throws an exception, and this task has not already
 * otherwise completed normally. On entry to this method, this task
 * {@link ForkJoinTask#isCompletedAbnormally}. The return value
 * of this method controls further propagation: If {@code true}
 * and this task has a completer that has not completed, then that
 * completer is also completed exceptionally, with the same
 * exception as this completer. The default implementation of
 * this method does nothing except return {@code true}.
 *
 * @param ex the exception
 * @param caller the task invoking this method (which may
 * be this task itself)
 * @return {@code true} if this exception should be propagated to this
 * task's completer, if one exists

```

```

*/
public boolean onExceptionalCompletion(Throwable ex, CountedCompleter<?> caller) {
    return true;
}

/**
 * Returns the completer established in this task's constructor,
 * or { @code null} if none.
 *
 * @return the completer
 */
public final CountedCompleter<?> getCompleter() {
    return completer;
}

/**
 * Returns the current pending count.
 *
 * @return the current pending count
 */
public final int getPendingCount() {
    return pending;
}

/**
 * Sets the pending count to the given value.
 *
 * @param count the count
 */
public final void setPendingCount(int count) {
    pending = count;
}

/**
 * Adds (atomically) the given value to the pending count.
 *
 * @param delta the value to add
 */
public final void addToPendingCount(int delta) {
    int c;
    do {} while (!U.compareAndSwapInt(this, PENDING, c = pending, c+delta));
}

/**
 * Sets (atomically) the pending count to the given count only if
 * it currently holds the given expected value.
 *
 * @param expected the expected value

```

```

* @param count the new value
* @return {@code true} if successful
*/
public final boolean compareAndSetPendingCount(int expected, int count) {
    return U.compareAndSwapInt(this, PENDING, expected, count);
}

/**
* If the pending count is nonzero, (atomically) decrements it.
*
* @return the initial (undecremented) pending count holding on entry
* to this method
*/
public final int decrementPendingCountUnlessZero() {
    int c;
    do {} while ((c = pending) != 0 &&
        !U.compareAndSwapInt(this, PENDING, c, c - 1));
    return c;
}

/**
* Returns the root of the current computation; i.e., this
* task if it has no completer, else its completer's root.
*
* @return the root of the current computation
*/
public final CountedCompleter<?> getRoot() {
    CountedCompleter<?> a = this, p;
    while ((p = a.completer) != null)
        a = p;
    return a;
}

/**
* If the pending count is nonzero, decrements the count;
* otherwise invokes {@link #onCompletion(CountedCompleter)}
* and then similarly tries to complete this task's completer,
* if one exists, else marks this task as complete.
*/
public final void tryComplete() {
    CountedCompleter<?> a = this, s = a;
    for (int c;;) {
        if ((c = a.pending) == 0) {
            a.onCompletion(s);
            if ((a = (s = a).completer) == null) {
                s.quietlyComplete();
                return;
            }
        }
    }
}

```

```

    }
    else if (U.compareAndSwapInt(a, PENDING, c, c - 1))
        return;
    }
}

/**
 * Equivalent to {@link #tryComplete} but does not invoke {@link
 * #onCompletion(CountedCompleter)} along the completion path:
 * If the pending count is nonzero, decrements the count;
 * otherwise, similarly tries to complete this task's completer, if
 * one exists, else marks this task as complete. This method may be
 * useful in cases where {@code onCompletion} should not, or need
 * not, be invoked for each completer in a computation.
 */
public final void propagateCompletion() {
    CountedCompleter<?> a = this, s = a;
    for (int c;;) {
        if ((c = a.pending) == 0) {
            if ((a = (s = a).completer) == null) {
                s.quietlyComplete();
                return;
            }
        }
        else if (U.compareAndSwapInt(a, PENDING, c, c - 1))
            return;
    }
}

/**
 * Regardless of pending count, invokes
 * {@link #onCompletion(CountedCompleter)}, marks this task as
 * complete and further triggers {@link #tryComplete} on this
 * task's completer, if one exists. The given rawResult is
 * used as an argument to {@link #setRawResult} before invoking
 * {@link #onCompletion(CountedCompleter)} or marking this task
 * as complete; its value is meaningful only for classes
 * overriding {@code setRawResult}. This method does not modify
 * the pending count.
 *
 * <p>This method may be useful when forcing completion as soon as
 * any one (versus all) of several subtask results are obtained.
 * However, in the common (and recommended) case in which {@code
 * setRawResult} is not overridden, this effect can be obtained
 * more simply using {@code quietlyCompleteRoot();}.
 *
 * @param rawResult the raw result
 */

```

```

public void complete(T rawResult) {
    CountedCompleter<?> p;
    setRawResult(rawResult);
    onCompletion(this);
    quietlyComplete();
    if ((p = completer) != null)
        p.tryComplete();
}

/**
 * If this task's pending count is zero, returns this task;
 * otherwise decrements its pending count and returns { @code
 * null }. This method is designed to be used with { @link
 * #nextComplete } in completion traversal loops.
 *
 * @return this task, if pending count was zero, else { @code null }
 */
public final CountedCompleter<?> firstComplete() {
    for (int c;;) {
        if ((c = pending) == 0)
            return this;
        else if (U.compareAndSwapInt(this, PENDING, c, c - 1))
            return null;
    }
}

/**
 * If this task does not have a completer, invokes { @link
 * ForkJoinTask#quietlyComplete } and returns { @code null }. Or, if
 * the completer's pending count is non-zero, decrements that
 * pending count and returns { @code null }. Otherwise, returns the
 * completer. This method can be used as part of a completion
 * traversal loop for homogeneous task hierarchies:
 *
 * <pre> { @code
 * for (CountedCompleter<?> c = firstComplete();
 *     c != null;
 *     c = c.nextComplete()) {
 * // ... process c ...
 * } }</pre>
 *
 * @return the completer, or { @code null } if none
 */
public final CountedCompleter<?> nextComplete() {
    CountedCompleter<?> p;
    if ((p = completer) != null)
        return p.firstComplete();
}

```



```

else {
    quietlyComplete();
    return null;
}
}

/**
 * Equivalent to { @code getRoot().quietlyComplete()}.
 */
public final void quietlyCompleteRoot() {
    for (CountedCompleter<?> a = this, p;;) {
        if ((p = a.completer) == null) {
            a.quietlyComplete();
            return;
        }
        a = p;
    }
}

/**
 * Supports ForkJoinTask exception propagation.
 */
void internalPropagateException(Throwable ex) {
    CountedCompleter<?> a = this, s = a;
    while (a.onExceptionalCompletion(ex, s) &&
        (a = (s = a).completer) != null && a.status >= 0 &&
        a.recordExceptionalCompletion(ex) == EXCEPTIONAL)
        ;
}

/**
 * Implements execution conventions for CountedCompleters.
 */
protected final boolean exec() {
    compute();
    return false;
}

/**
 * Returns the result of the computation. By default
 * returns { @code null}, which is appropriate for { @code Void}
 * actions, but in other cases should be overridden, almost
 * always to return a field or function of a field that
 * holds the result upon completion.
 *
 * @return the result of the computation
 */
public T getRawResult() { return null; }

```

```

/**
 * A method that result-bearing CountedCompleters may optionally
 * use to help maintain result data. By default, does nothing.
 * Overrides are not recommended. However, if this method is
 * overridden to update existing objects or fields, then it must
 * in general be defined to be thread-safe.
 */
protected void setRawResult(T t) { }

// Unsafe mechanics
private static final sun.misc.Unsafe U;
private static final long PENDING;
static {
    try {
        U = getUnsafe();
        PENDING = U.objectFieldOffset
            (CountedCompleter.class.getDeclaredField("pending"));
    } catch (Exception e) {
        throw new Error(e);
    }
}

/**
 * Returns a sun.misc.Unsafe. Suitable for use in a 3rd party package.
 * Replace with a simple call to Unsafe.getUnsafe when integrating
 * into a jdk.
 *
 * @return a sun.misc.Unsafe
 */
private static sun.misc.Unsafe getUnsafe() {
    try {
        return sun.misc.Unsafe.getUnsafe();
    } catch (SecurityException tryReflectionInstead) {}
    try {
        return java.security.AccessController.doPrivileged
            (new java.security.PrivilegedExceptionAction<sun.misc.Unsafe>() {
                public sun.misc.Unsafe run() throws Exception {
                    Class<sun.misc.Unsafe> k = sun.misc.Unsafe.class;
                    for (java.lang.reflect.Field f : k.getDeclaredFields()) {
                        f.setAccessible(true);
                        Object x = f.get(null);
                        if (k.isInstance(x))
                            return k.cast(x);
                    }
                    throw new NoSuchFieldError("the Unsafe");
                }
            });
    } catch (java.security.PrivilegedActionException e) {

```

```

        throw new RuntimeException("Could not initialize intrinsics",
            e.getCause());
    }
}
}
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

package io.netty.util.concurrent;

import java.util.concurrent.TimeUnit;

/**
 * A skeletal {@link Future} implementation which represents a {@link Future} which has been completed already.
 */
public abstract class CompleteFuture<V> extends AbstractFuture<V> {

    private final EventExecutor executor;

    /**
     * Creates a new instance.
     *
     * @param executor the {@link EventExecutor} associated with this future
     */
    protected CompleteFuture(EventExecutor executor) {
        this.executor = executor;
    }

    /**
     * Return the {@link EventExecutor} which is used by this {@link CompleteFuture}.
     */
    protected EventExecutor executor() {
        return executor;
    }
}

```

```

@Override
public Future<V> addListener(GenericFutureListener<? extends Future<? super V>> listener) {
    if (listener == null) {
        throw new NullPointerException("listener");
    }
    DefaultPromise.notifyListener(executor(), this, listener);
    return this;
}

```

```

@Override
public Future<V> addListeners(GenericFutureListener<? extends Future<? super V>>... listeners) {
    if (listeners == null) {
        throw new NullPointerException("listeners");
    }
    for (GenericFutureListener<? extends Future<? super V>> l: listeners) {
        if (l == null) {
            break;
        }
        DefaultPromise.notifyListener(executor(), this, l);
    }
    return this;
}

```

```

@Override
public Future<V> removeListener(GenericFutureListener<? extends Future<? super V>> listener) {
    // NOOP
    return this;
}

```

```

@Override
public Future<V> removeListeners(GenericFutureListener<? extends Future<? super V>>... listeners) {
    // NOOP
    return this;
}

```

```

@Override
public Future<V> await() throws InterruptedException {
    if (Thread.interrupted()) {
        throw new InterruptedException();
    }
    return this;
}

```

```

@Override
public boolean await(long timeout, TimeUnit unit) throws InterruptedException {
    if (Thread.interrupted()) {
        throw new InterruptedException();
    }
}

```

```

    return true;
}

@Override
public Future<V> sync() throws InterruptedException {
    return this;
}

@Override
public Future<V> syncUninterruptibly() {
    return this;
}

@Override
public boolean await(long timeoutMillis) throws InterruptedException {
    if (Thread.interrupted()) {
        throw new InterruptedException();
    }
    return true;
}

@Override
public Future<V> awaitUninterruptibly() {
    return this;
}

@Override
public boolean awaitUninterruptibly(long timeout, TimeUnit unit) {
    return true;
}

@Override
public boolean awaitUninterruptibly(long timeoutMillis) {
    return true;
}

@Override
public boolean isDone() {
    return true;
}

@Override
public boolean isCancelled() {
    return false;
}

@Override
public boolean isCancelled() {

```

```

    return false;
}

@Override
public boolean cancel(boolean mayInterruptIfRunning) {
    return false;
}
}
/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */

package io.netty.util;

/**
 * An {@link IllegalStateException} which is raised when a user attempts to access a {@link ReferenceCounted}
 * whose
 * reference count has been decreased to 0 (and consequently freed).
 */
public class IllegalReferenceCountException extends IllegalStateException {

    private static final long serialVersionUID = -2507492394288153468L;

    public IllegalReferenceCountException() { }

    public IllegalReferenceCountException(int refCnt) {
        this("refCnt: " + refCnt);
    }

    public IllegalReferenceCountException(int refCnt, int increment) {
        this("refCnt: " + refCnt + ", " + (increment > 0 ? "increment: " + increment : "decrement: " + -increment));
    }

    public IllegalReferenceCountException(String message) {
        super(message);
    }
}

```

```

public IllegalReferenceCountException(String message, Throwable cause) {
    super(message, cause);
}

public IllegalReferenceCountException(Throwable cause) {
    super(cause);
}
}
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
package io.netty.channel;

import io.netty.util.ReferenceCountUtil;
import io.netty.util.internal.TypeParameterMatcher;

/**
 * {@link ChannelInboundHandlerAdapter} which allows to explicit only handle a specific type of messages.
 *
 * For example here is an implementation which only handle {@link String} messages.
 *
 * <pre>
 * public class StringHandler extends
 *     {@link SimpleChannelInboundHandler}&lt;{@link String}&gt; {
 *
 *     {@code @Override}
 *     protected void channelRead0({@link ChannelHandlerContext} ctx, {@link String} message)
 *         throws {@link Exception} {
 *         System.out.println(message);
 *     }
 * }
 * </pre>
 *
 * Be aware that depending of the constructor parameters it will release all handled messages.
 */

```

\* <h3>Forward compatibility notice</h3>

\* <p>

\* Please keep in mind that { @link #channelRead0(ChannelHandlerContext, Object)} will be renamed to

\* { @code messageReceived(ChannelHandlerContext, I)} in 5.0.

\* </p>

\*/

```
public abstract class SimpleChannelInboundHandler<I> extends ChannelInboundHandlerAdapter {
```

```
    private final TypeParameterMatcher matcher;
```

```
    private final boolean autoRelease;
```

```
    /**
```

```
     * @see { @link #SimpleChannelInboundHandler(boolean)} with { @code true} as boolean parameter.
```

```
    */
```

```
    protected SimpleChannelInboundHandler() {
```

```
        this(true);
```

```
    }
```

```
    /**
```

```
     * Create a new instance which will try to detect the types to match out of the type parameter of the class.
```

```
     *
```

```
     * @param autoRelease { @code true} if handled messages should be released automatically by pass them to
```

```
     *     { @link ReferenceCountUtil#release(Object)}.
```

```
    */
```

```
    protected SimpleChannelInboundHandler(boolean autoRelease) {
```

```
        matcher = TypeParameterMatcher.find(this, SimpleChannelInboundHandler.class, "I");
```

```
        this.autoRelease = autoRelease;
```

```
    }
```

```
    /**
```

```
     * @see { @link #SimpleChannelInboundHandler(Class, boolean)} with { @code true} as boolean value.
```

```
    */
```

```
    protected SimpleChannelInboundHandler(Class<? extends I> inboundMessageType) {
```

```
        this(inboundMessageType, true);
```

```
    }
```

```
    /**
```

```
     * Create a new instance
```

```
     *
```

```
     * @param inboundMessageType The type of messages to match
```

```
     * @param autoRelease { @code true} if handled messages should be released automatically by pass them to
```

```
     *     { @link ReferenceCountUtil#release(Object)}.
```

```
    */
```

```
    protected SimpleChannelInboundHandler(Class<? extends I> inboundMessageType, boolean autoRelease) {
```

```
        matcher = TypeParameterMatcher.get(inboundMessageType);
```

```
        this.autoRelease = autoRelease;
```

```
    }
```



```

/**
 * Returns {@code true} if the given message should be handled. If {@code false} it will be passed to the next
 * {@link ChannelInboundHandler} in the {@link ChannelPipeline}.
 */
public boolean acceptInboundMessage(Object msg) throws Exception {
    return matcher.match(msg);
}

@Override
public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
    boolean release = true;
    try {
        if (acceptInboundMessage(msg)) {
            @SuppressWarnings("unchecked")
            I imsg = (I) msg;
            channelRead0(ctx, imsg);
        } else {
            release = false;
            ctx.fireChannelRead(msg);
        }
    } finally {
        if (autoRelease && release) {
            ReferenceCountUtil.release(msg);
        }
    }
}

/**
 * Please keep in mind that this method will be renamed to
 * {@code messageReceived(ChannelHandlerContext, I)} in 5.0.
 *
 * Is called for each message of type {@link I}.
 *
 * @param ctx the {@link ChannelHandlerContext} which this {@link SimpleChannelInboundHandler}
 * belongs to
 * @param msg the message to handle
 * @throws Exception is thrown if an error occurred
 */
protected abstract void channelRead0(ChannelHandlerContext ctx, I msg) throws Exception;
}

/**
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0

```

```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
package io.netty.channel;

import io.netty.util.concurrent.CompleteFuture;
import io.netty.util.concurrent.EventExecutor;
import io.netty.util.concurrent.Future;
import io.netty.util.concurrent.GenericFutureListener;

/**
 * A skeletal {@link ChannelFuture} implementation which represents a
 * {@link ChannelFuture} which has been completed already.
 */
abstract class CompleteChannelFuture extends CompleteFuture<Void> implements ChannelFuture {

    private final Channel channel;

    /**
     * Creates a new instance.
     *
     * @param channel the {@link Channel} associated with this future
     */
    protected CompleteChannelFuture(Channel channel, EventExecutor executor) {
        super(executor);
        if (channel == null) {
            throw new NullPointerException("channel");
        }
        this.channel = channel;
    }

    @Override
    protected EventExecutor executor() {
        EventExecutor e = super.executor();
        if (e == null) {
            return channel().eventLoop();
        } else {
            return e;
        }
    }

    @Override
    public ChannelFuture addListener(GenericFutureListener<? extends Future<? super Void>> listener) {
        super.addListener(listener);
    }

```

```

    return this;
}

@Override
public ChannelFuture addListeners(GenericFutureListener<? extends Future<? super Void>>... listeners) {
    super.addListeners(listeners);
    return this;
}

@Override
public ChannelFuture removeListener(GenericFutureListener<? extends Future<? super Void>> listener) {
    super.removeListener(listener);
    return this;
}

@Override
public ChannelFuture removeListeners(GenericFutureListener<? extends Future<? super Void>>... listeners) {
    super.removeListeners(listeners);
    return this;
}

@Override
public ChannelFuture syncUninterruptibly() {
    return this;
}

@Override
public ChannelFuture sync() throws InterruptedException {
    return this;
}

@Override
public ChannelFuture await() throws InterruptedException {
    return this;
}

@Override
public ChannelFuture awaitUninterruptibly() {
    return this;
}

@Override
public Channel channel() {
    return channel;
}

@Override
public Void getNow() {

```

```

    return null;
}
}
/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
package io.netty.buffer;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.channels.GatheringByteChannel;
import java.nio.channels.ScatteringByteChannel;

/**
 * A derived buffer which simply forwards all data access requests to its
 * parent. It is recommended to use {@link ByteBuffer#duplicate()} instead
 * of calling the constructor explicitly.
 */
public class DuplicatedByteBuffer extends AbstractDerivedByteBuffer {

    private final ByteBuffer buffer;

    public DuplicatedByteBuffer(ByteBuffer buffer) {
        super(buffer.maxCapacity());

        if (buffer instanceof DuplicatedByteBuffer) {
            this.buffer = ((DuplicatedByteBuffer) buffer).buffer;
        } else {
            this.buffer = buffer;
        }

        setIndex(buffer.readerIndex(), buffer.writerIndex());
    }

```

```

}

@Override
public ByteBuf unwrap() {
    return buffer;
}

@Override
public ByteBufAllocator alloc() {
    return buffer.alloc();
}

@Override
public ByteOrder order() {
    return buffer.order();
}

@Override
public boolean isDirect() {
    return buffer.isDirect();
}

@Override
public int capacity() {
    return buffer.capacity();
}

@Override
public ByteBuf capacity(int newCapacity) {
    buffer.capacity(newCapacity);
    return this;
}

@Override
public boolean hasArray() {
    return buffer.hasArray();
}

@Override
public byte[] array() {
    return buffer.array();
}

@Override
public int arrayOffset() {
    return buffer.arrayOffset();
}

```

```
@Override
public boolean hasMemoryAddress() {
    return buffer.hasMemoryAddress();
}
```

```
@Override
public long memoryAddress() {
    return buffer.memoryAddress();
}
```

```
@Override
public byte getByte(int index) {
    return _getByte(index);
}
```

```
@Override
protected byte _getByte(int index) {
    return buffer.getByte(index);
}
```

```
@Override
public short getShort(int index) {
    return _getShort(index);
}
```

```
@Override
protected short _getShort(int index) {
    return buffer.getShort(index);
}
```

```
@Override
public int getUnsignedMedium(int index) {
    return _getUnsignedMedium(index);
}
```

```
@Override
protected int _getUnsignedMedium(int index) {
    return buffer.getUnsignedMedium(index);
}
```

```
@Override
public int getInt(int index) {
    return _getInt(index);
}
```

```
@Override
protected int _getInt(int index) {
    return buffer.getInt(index);
}
```

```

}

@Override
public long getLong(int index) {
    return _getLong(index);
}

@Override
protected long _getLong(int index) {
    return buffer.getLong(index);
}

@Override
public ByteBuf copy(int index, int length) {
    return buffer.copy(index, length);
}

@Override
public ByteBuf slice(int index, int length) {
    return buffer.slice(index, length);
}

@Override
public ByteBuf getBytes(int index, ByteBuf dst, int dstIndex, int length) {
    buffer.getBytes(index, dst, dstIndex, length);
    return this;
}

@Override
public ByteBuf getBytes(int index, byte[] dst, int dstIndex, int length) {
    buffer.getBytes(index, dst, dstIndex, length);
    return this;
}

@Override
public ByteBuf getBytes(int index, ByteBuffer dst) {
    buffer.getBytes(index, dst);
    return this;
}

@Override
public ByteBuf setByte(int index, int value) {
    _setByte(index, value);
    return this;
}

@Override
protected void _setByte(int index, int value) {

```

```

        buffer.setByte(index, value);
    }

    @Override
    public ByteBuf setShort(int index, int value) {
        _setShort(index, value);
        return this;
    }

    @Override
    protected void _setShort(int index, int value) {
        buffer.setShort(index, value);
    }

    @Override
    public ByteBuf setMedium(int index, int value) {
        _setMedium(index, value);
        return this;
    }

    @Override
    protected void _setMedium(int index, int value) {
        buffer.setMedium(index, value);
    }

    @Override
    public ByteBuf setInt(int index, int value) {
        _setInt(index, value);
        return this;
    }

    @Override
    protected void _setInt(int index, int value) {
        buffer.setInt(index, value);
    }

    @Override
    public ByteBuf setLong(int index, long value) {
        _setLong(index, value);
        return this;
    }

    @Override
    protected void _setLong(int index, long value) {
        buffer.setLong(index, value);
    }

    @Override

```



```

public ByteBuf setBytes(int index, byte[] src, int srcIndex, int length) {
    buffer.setBytes(index, src, srcIndex, length);
    return this;
}

@Override
public ByteBuf setBytes(int index, ByteBuf src, int srcIndex, int length) {
    buffer.setBytes(index, src, srcIndex, length);
    return this;
}

@Override
public ByteBuf setBytes(int index, ByteBuffer src) {
    buffer.setBytes(index, src);
    return this;
}

@Override
public ByteBuf getBytes(int index, OutputStream out, int length)
    throws IOException {
    buffer.getBytes(index, out, length);
    return this;
}

@Override
public int getBytes(int index, GatheringByteChannel out, int length)
    throws IOException {
    return buffer.getBytes(index, out, length);
}

@Override
public int setBytes(int index, InputStream in, int length)
    throws IOException {
    return buffer.setBytes(index, in, length);
}

@Override
public int setBytes(int index, ScatteringByteChannel in, int length)
    throws IOException {
    return buffer.setBytes(index, in, length);
}

@Override
public int nioBufferCount() {
    return buffer.nioBufferCount();
}

@Override

```

```

public ByteBuffer[] nioBuffers(int index, int length) {
    return buffer.nioBuffers(index, length);
}

@Override
public ByteBuffer internalNioBuffer(int index, int length) {
    return nioBuffer(index, length);
}

@Override
public int forEachByte(int index, int length, ByteBufProcessor processor) {
    return buffer.forEachByte(index, length, processor);
}

@Override
public int forEachByteDesc(int index, int length, ByteBufProcessor processor) {
    return buffer.forEachByteDesc(index, length, processor);
}
}
}
/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
package io.netty.buffer;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.channels.GatheringByteChannel;
import java.nio.channels.ScatteringByteChannel;

/**
 * A derived buffer which exposes its parent's sub-region only. It is
 * recommended to use {@link ByteBuf#slice()} and

```

```

* {@link ByteBuf#slice(int, int)} instead of calling the constructor
* explicitly.
*/
public class SlicedByteBuf extends AbstractDerivedByteBuf {

    private final ByteBuf buffer;
    private final int adjustment;
    private final int length;

    public SlicedByteBuf(ByteBuf buffer, int index, int length) {
        super(length);
        if (index < 0 || index > buffer.capacity() - length) {
            throw new IndexOutOfBoundsException(buffer.toString() + ".slice(" + index + ", " + length + ')');
        }

        if (buffer instanceof SlicedByteBuf) {
            this.buffer = ((SlicedByteBuf) buffer).buffer;
            adjustment = ((SlicedByteBuf) buffer).adjustment + index;
        } else if (buffer instanceof DuplicatedByteBuf) {
            this.buffer = buffer.unwrap();
            adjustment = index;
        } else {
            this.buffer = buffer;
            adjustment = index;
        }
        this.length = length;

        writerIndex(length);
    }

    @Override
    public ByteBuf unwrap() {
        return buffer;
    }

    @Override
    public ByteBufAllocator alloc() {
        return buffer.alloc();
    }

    @Override
    public ByteOrder order() {
        return buffer.order();
    }

    @Override
    public boolean isDirect() {
        return buffer.isDirect();
    }

```

```

}

@Override
public int capacity() {
    return length;
}

@Override
public ByteBuf capacity(int newCapacity) {
    throw new UnsupportedOperationException("sliced buffer");
}

@Override
public boolean hasArray() {
    return buffer.hasArray();
}

@Override
public byte[] array() {
    return buffer.array();
}

@Override
public int arrayOffset() {
    return buffer.arrayOffset() + adjustment;
}

@Override
public boolean hasMemoryAddress() {
    return buffer.hasMemoryAddress();
}

@Override
public long memoryAddress() {
    return buffer.memoryAddress() + adjustment;
}

@Override
protected byte _getBytes(int index) {
    return buffer.getBytes(index + adjustment);
}

@Override
protected short _getShort(int index) {
    return buffer.getShort(index + adjustment);
}

@Override

```

```

protected int _getUnsignedMedium(int index) {
    return buffer.getUnsignedMedium(index + adjustment);
}

@Override
protected int _getInt(int index) {
    return buffer.getInt(index + adjustment);
}

@Override
protected long _getLong(int index) {
    return buffer.getLong(index + adjustment);
}

@Override
public ByteBuffer duplicate() {
    ByteBuffer duplicate = buffer.slice(adjustment, length);
    duplicate.setIndex(readerIndex(), writerIndex());
    return duplicate;
}

@Override
public ByteBuffer copy(int index, int length) {
    checkIndex(index, length);
    return buffer.copy(index + adjustment, length);
}

@Override
public ByteBuffer slice(int index, int length) {
    checkIndex(index, length);
    if (length == 0) {
        return Unpooled.EMPTY_BUFFER;
    }
    return buffer.slice(index + adjustment, length);
}

@Override
public ByteBuffer getBytes(int index, ByteBuffer dst, int dstIndex, int length) {
    checkIndex(index, length);
    buffer.getBytes(index + adjustment, dst, dstIndex, length);
    return this;
}

@Override
public ByteBuffer getBytes(int index, byte[] dst, int dstIndex, int length) {
    checkIndex(index, length);
    buffer.getBytes(index + adjustment, dst, dstIndex, length);
    return this;
}

```

```

}

@Override
public ByteBuffer getBytes(int index, ByteBuffer dst) {
    checkIndex(index, dst.remaining());
    buffer.getBytes(index + adjustment, dst);
    return this;
}

@Override
protected void _setByte(int index, int value) {
    buffer.setByte(index + adjustment, value);
}

@Override
protected void _setShort(int index, int value) {
    buffer.setShort(index + adjustment, value);
}

@Override
protected void _setMedium(int index, int value) {
    buffer.setMedium(index + adjustment, value);
}

@Override
protected void _setInt(int index, int value) {
    buffer.setInt(index + adjustment, value);
}

@Override
protected void _setLong(int index, long value) {
    buffer.setLong(index + adjustment, value);
}

@Override
public ByteBuffer setBytes(int index, byte[] src, int srcIndex, int length) {
    checkIndex(index, length);
    buffer.setBytes(index + adjustment, src, srcIndex, length);
    return this;
}

@Override
public ByteBuffer setBytes(int index, ByteBuffer src, int srcIndex, int length) {
    checkIndex(index, length);
    buffer.setBytes(index + adjustment, src, srcIndex, length);
    return this;
}

```

```

@Override
public ByteBuf setBytes(int index, ByteBuffer src) {
    checkIndex(index, src.remaining());
    buffer.setBytes(index + adjustment, src);
    return this;
}

@Override
public ByteBuf getBytes(int index, OutputStream out, int length) throws IOException {
    checkIndex(index, length);
    buffer.getBytes(index + adjustment, out, length);
    return this;
}

@Override
public int getBytes(int index, GatheringByteChannel out, int length) throws IOException {
    checkIndex(index, length);
    return buffer.getBytes(index + adjustment, out, length);
}

@Override
public int setBytes(int index, InputStream in, int length) throws IOException {
    checkIndex(index, length);
    return buffer.setBytes(index + adjustment, in, length);
}

@Override
public int setBytes(int index, ScatteringByteChannel in, int length) throws IOException {
    checkIndex(index, length);
    return buffer.setBytes(index + adjustment, in, length);
}

@Override
public int.nioBufferCount() {
    return buffer.nioBufferCount();
}

@Override
public ByteBuffer.nioBuffer(int index, int length) {
    checkIndex(index, length);
    return buffer.nioBuffer(index + adjustment, length);
}

@Override
public ByteBuffer[].nioBuffers(int index, int length) {
    checkIndex(index, length);
    return buffer.nioBuffers(index + adjustment, length);
}

```

```

@Override
public ByteBuffer internalNioBuffer(int index, int length) {
    checkIndex(index, length);
    return nioBuffer(index, length);
}

@Override
public int forEachByte(int index, int length, ByteBufProcessor processor) {
    int ret = buffer.forEachByte(index + adjustment, length, processor);
    if (ret >= adjustment) {
        return ret - adjustment;
    } else {
        return -1;
    }
}

@Override
public int forEachByteDesc(int index, int length, ByteBufProcessor processor) {
    int ret = buffer.forEachByteDesc(index + adjustment, length, processor);
    if (ret >= adjustment) {
        return ret - adjustment;
    } else {
        return -1;
    }
}
}
}
/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
package io.netty.handler.codec.marshalling;

import org.jboss.marshalling.ByteInput;

import java.io.IOException;

```



```

/**
 * {@link ByteArrayInputStream} implementation which wraps another {@link ByteArrayInputStream} and throws a {@link
 * TooBigObjectException}
 * if the read limit was reached.
 */
class LimitingByteInput implements ByteArrayInputStream {

    // Use a static instance here to remove the overhead of fillStackTrace
    private static final TooBigObjectException EXCEPTION = new TooBigObjectException();

    private final ByteArrayInputStream input;
    private final long limit;
    private long read;

    public LimitingByteInput(ByteArrayInputStream input, long limit) {
        if (limit <= 0) {
            throw new IllegalArgumentException("The limit MUST be > 0");
        }
        this.input = input;
        this.limit = limit;
    }

    @Override
    public void close() throws IOException {
        // Nothing to do
    }

    @Override
    public int available() throws IOException {
        return readable(input.available());
    }

    @Override
    public int read() throws IOException {
        int readable = readable(1);
        if (readable > 0) {
            int b = input.read();
            read++;
            return b;
        } else {
            throw EXCEPTION;
        }
    }

    @Override
    public int read(byte[] array) throws IOException {
        return read(array, 0, array.length);
    }
}

```

```

@Override
public int read(byte[] array, int offset, int length) throws IOException {
    int readable = readable(length);
    if (readable > 0) {
        int i = input.read(array, offset, readable);
        read += i;
        return i;
    } else {
        throw EXCEPTION;
    }
}

@Override
public long skip(long bytes) throws IOException {
    int readable = readable((int) bytes);
    if (readable > 0) {
        long i = input.skip(readable);
        read += i;
        return i;
    } else {
        throw EXCEPTION;
    }
}

private int readable(int length) {
    return (int) Math.min(length, limit - read);
}

/**
 * Exception that will get thrown if the {@link Object} is too big to unmarshall
 *
 */
static final class TooBigObjectException extends IOException {
    private static final long serialVersionUID = 1L;
}

/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT

```

- \* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
- \* License for the specific language governing permissions and limitations
- \* under the License.
- \*/

```
package io.netty.handler.codec.sctp;
```

```
import io.netty.buffer.ByteBuf;
import io.netty.buffer.Unpooled;
import io.netty.channel.ChannelHandlerContext;
import io.netty.channel.ChannelInboundHandler;
import io.netty.channel.sctp.SctpMessage;
import io.netty.handler.codec.MessageToMessageDecoder;
```

```
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
/**
 * {@link MessageToMessageDecoder} which will take care of handle fragmented {@link SctpMessage}s, so
 * only <strong>complete</strong> {@link SctpMessage}s will be forwarded to the next
 * {@link ChannelInboundHandler}.
 */
```

```
public class SctpMessageCompletionHandler extends MessageToMessageDecoder<SctpMessage> {
    private final Map<Integer, ByteBuf> fragments = new HashMap<Integer, ByteBuf>();
```

```
    @Override
```

```
    protected void decode(ChannelHandlerContext ctx, SctpMessage msg, List<Object> out) throws Exception {
        final ByteBuf byteBuf = msg.content();
        final int protocolIdentifier = msg.protocolIdentifier();
        final int streamIdentifier = msg.streamIdentifier();
        final boolean isComplete = msg.isComplete();
```

```
        ByteBuf frag;
        if (fragments.containsKey(streamIdentifier)) {
            frag = fragments.remove(streamIdentifier);
        } else {
            frag = Unpooled.EMPTY_BUFFER;
        }
```

```
        if (isComplete && !frag.isReadable()) {
            //data chunk is not fragmented
            out.add(msg);
        } else if (!isComplete && frag.isReadable()) {
            //more message to complete
            fragments.put(streamIdentifier, Unpooled.wrappedBuffer(frag, byteBuf));
        } else if (isComplete && frag.isReadable()) {
            //last message to complete
```

```

        fragments.remove(streamIdentifier);
        SctpMessage assembledMsg = new SctpMessage(
            protocolIdentifier,
            streamIdentifier,
            Unpooled.wrappedBuffer(frag, byteBuf));
        out.add(assembledMsg);
    } else {
        //first incomplete message
        fragments.put(streamIdentifier, byteBuf);
    }
    byteBuf.retain();
}
}
/*
 * Copyright 2012 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
package io.netty.handler.codec;

import io.netty.buffer.ByteBuf;
import io.netty.buffer.Unpooled;

/**
 * A set of commonly used delimiters for {@link DelimiterBasedFrameDecoder}.
 */
public final class Delimiters {

    /**
     * Returns a {@code NUL (0x00)} delimiter, which could be used for
     * Flash XML socket or any similar protocols.
     */
    public static ByteBuf[] nulDelimiter() {
        return new ByteBuf[] {
            Unpooled.wrappedBuffer(new byte[] { 0 });
        }
    }

    /**

```

```

* Returns {@code CR ('\r')} and {@code LF ('\n')} delimiters, which could
* be used for text-based line protocols.
*/
public static ByteBuf[] lineDelimiter() {
    return new ByteBuf[] {
        Unpooled.wrappedBuffer(new byte[] { '\r', '\n' }),
        Unpooled.wrappedBuffer(new byte[] { '\n' }),
    };
}

private Delimiters() {
    // Unused
}
}
/*
* Copyright 2012 The Netty Project
*
* The Netty Project licenses this file to you under the Apache License,
* version 2.0 (the "License"); you may not use this file except in compliance
* with the License. You may obtain a copy of the License at:
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
* WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
* License for the specific language governing permissions and limitations
* under the License.
*/
package io.netty.handler.codec;

import io.netty.buffer.ByteBuf;
import io.netty.channel.ChannelHandlerContext;

import java.util.List;

/**
* A decoder that splits the received {@link ByteBuf}s by one or more
* delimiters. It is particularly useful for decoding the frames which ends
* with a delimiter such as {@link Delimiters#nulDelimiter() NUL} or
* {@link Delimiters#lineDelimiter() newline characters}.
*
* 

### Predefined delimiters


* 

* {@link Delimiters} defines frequently used delimiters for convenience' sake.
*
* 

### Specifying more than one delimiter


*


```

```

* {@link DelimiterBasedFrameDecoder} allows you to specify more than one
* delimiter. If more than one delimiter is found in the buffer, it chooses
* the delimiter which produces the shortest frame. For example, if you have
* the following data in the buffer:
* <pre>
* +-----+
* | ABC\nDEF\r\n |
* +-----+
* </pre>
* a {@link DelimiterBasedFrameDecoder}({@link Delimiters#lineDelimiter() Delimiters.lineDelimiter()})
* will choose {@code '\n'} as the first delimiter and produce two frames:
* <pre>
* +----+----+
* | ABC | DEF |
* +----+----+
* </pre>
* rather than incorrectly choosing {@code '\r\n'} as the first delimiter:
* <pre>
* +-----+
* | ABC\nDEF |
* +-----+
* </pre>
*/

```

```

public class DelimiterBasedFrameDecoder extends ByteToMessageDecoder {

    private final ByteBuf[] delimiters;
    private final int maxFrameLength;
    private final boolean stripDelimiter;
    private final boolean failFast;
    private boolean discardingTooLongFrame;
    private int tooLongFrameLength;
    /** Set only when decoding with "\n" and "\r\n" as the delimiter. */
    private final LineBasedFrameDecoder lineBasedDecoder;

    /**
     * Creates a new instance.
     *
     * @param maxFrameLength the maximum length of the decoded frame.
     *                        A {@link TooLongFrameException} is thrown if
     *                        the length of the frame exceeds this value.
     * @param delimiter the delimiter
     */
    public DelimiterBasedFrameDecoder(int maxFrameLength, ByteBuf delimiter) {
        this(maxFrameLength, true, delimiter);
    }

    /**
     * Creates a new instance.
     */

```

```

*
* @param maxFrameLength the maximum length of the decoded frame.
*     A { @link TooLongFrameException } is thrown if
*     the length of the frame exceeds this value.
* @param stripDelimiter whether the decoded frame should strip out the
*     delimiter or not
* @param delimiter the delimiter
*/
public DelimiterBasedFrameDecoder(
    int maxFrameLength, boolean stripDelimiter, ByteBuf delimiter) {
    this(maxFrameLength, stripDelimiter, true, delimiter);
}

/**
 * Creates a new instance.
 *
 * @param maxFrameLength the maximum length of the decoded frame.
 *     A { @link TooLongFrameException } is thrown if
 *     the length of the frame exceeds this value.
 * @param stripDelimiter whether the decoded frame should strip out the
 *     delimiter or not
 * @param failFast If <tt>true</tt>, a { @link TooLongFrameException } is
 *     thrown as soon as the decoder notices the length of the
 *     frame will exceed <tt>maxFrameLength</tt> regardless of
 *     whether the entire frame has been read.
 *     If <tt>false</tt>, a { @link TooLongFrameException } is
 *     thrown after the entire frame that exceeds
 *     <tt>maxFrameLength</tt> has been read.
 * @param delimiter the delimiter
 */
public DelimiterBasedFrameDecoder(
    int maxFrameLength, boolean stripDelimiter, boolean failFast,
    ByteBuf delimiter) {
    this(maxFrameLength, stripDelimiter, failFast, new ByteBuf[] {
        delimiter.slice(delimiter.readerIndex(), delimiter.readableBytes())});
}

/**
 * Creates a new instance.
 *
 * @param maxFrameLength the maximum length of the decoded frame.
 *     A { @link TooLongFrameException } is thrown if
 *     the length of the frame exceeds this value.
 * @param delimiters the delimiters
 */
public DelimiterBasedFrameDecoder(int maxFrameLength, ByteBuf... delimiters) {
    this(maxFrameLength, true, delimiters);
}

```

```

/**
 * Creates a new instance.
 *
 * @param maxFrameLength the maximum length of the decoded frame.
 *       A { @link TooLongFrameException } is thrown if
 *       the length of the frame exceeds this value.
 * @param stripDelimiter whether the decoded frame should strip out the
 *       delimiter or not
 * @param delimiters the delimiters
 */
public DelimiterBasedFrameDecoder(
    int maxFrameLength, boolean stripDelimiter, ByteBuf... delimiters) {
    this(maxFrameLength, stripDelimiter, true, delimiters);
}

/**
 * Creates a new instance.
 *
 * @param maxFrameLength the maximum length of the decoded frame.
 *       A { @link TooLongFrameException } is thrown if
 *       the length of the frame exceeds this value.
 * @param stripDelimiter whether the decoded frame should strip out the
 *       delimiter or not
 * @param failFast If <tt>true</tt>, a { @link TooLongFrameException } is
 *       thrown as soon as the decoder notices the length of the
 *       frame will exceed <tt>maxFrameLength</tt> regardless of
 *       whether the entire frame has been read.
 *       If <tt>>false</tt>, a { @link TooLongFrameException } is
 *       thrown after the entire frame that exceeds
 *       <tt>maxFrameLength</tt> has been read.
 * @param delimiters the delimiters
 */
public DelimiterBasedFrameDecoder(
    int maxFrameLength, boolean stripDelimiter, boolean failFast, ByteBuf... delimiters) {
    validateMaxFrameLength(maxFrameLength);
    if (delimiters == null) {
        throw new NullPointerException("delimiters");
    }
    if (delimiters.length == 0) {
        throw new IllegalArgumentException("empty delimiters");
    }

    if (isLineBased(delimiters) && !isSubclass()) {
        lineBasedDecoder = new LineBasedFrameDecoder(maxFrameLength, stripDelimiter, failFast);
        this.delimiters = null;
    } else {
        this.delimiters = new ByteBuf[delimiters.length];
    }
}

```



```

        for (int i = 0; i < delimiters.length; i++) {
            ByteBuf d = delimiters[i];
            validateDelimiter(d);
            this.delimiters[i] = d.slice(d.readerIndex(), d.readableBytes());
        }
        lineBasedDecoder = null;
    }
    this.maxFrameLength = maxFrameLength;
    this.stripDelimiter = stripDelimiter;
    this.failFast = failFast;
}

/** Returns true if the delimiters are "\n" and "\r\n". */
private static boolean isLineBased(final ByteBuf[] delimiters) {
    if (delimiters.length != 2) {
        return false;
    }
    ByteBuf a = delimiters[0];
    ByteBuf b = delimiters[1];
    if (a.capacity() < b.capacity()) {
        a = delimiters[1];
        b = delimiters[0];
    }
    return a.capacity() == 2 && b.capacity() == 1
        && a.getBytes(0) == '\r' && a.getBytes(1) == '\n'
        && b.getBytes(0) == '\n';
}

/**
 * Return {@code true} if the current instance is a subclass of DelimiterBasedFrameDecoder
 */
private boolean isSubclass() {
    return getClass() != DelimiterBasedFrameDecoder.class;
}

@Override
protected final void decode(ChannelHandlerContext ctx, ByteBuf in, List<Object> out) throws Exception {
    Object decoded = decode(ctx, in);
    if (decoded != null) {
        out.add(decoded);
    }
}

/**
 * Create a frame out of the {@link ByteBuf} and return it.
 *
 * @param ctx the {@link ChannelHandlerContext} which this {@link ByteToMessageDecoder}
belongs to

```

```

* @param buffer    the {@link ByteBuffer} from which to read data
* @return frame    the {@link ByteBuffer} which represent the frame or {@code null} if no frame could
*                  be created.
*/

```

```
protected Object decode(ChannelHandlerContext ctx, ByteBuffer buffer) throws Exception {
```

```

    if (lineBasedDecoder != null) {
        return lineBasedDecoder.decode(ctx, buffer);
    }
    // Try all delimiters and choose the delimiter which yields the shortest frame.
    int minFrameLength = Integer.MAX_VALUE;
    ByteBuffer minDelim = null;
    for (ByteBuffer delim: delimiters) {
        int frameLength = indexOf(buffer, delim);
        if (frameLength >= 0 && frameLength < minFrameLength) {
            minFrameLength = frameLength;
            minDelim = delim;
        }
    }
}

```

```

if (minDelim != null) {
    int minDelimLength = minDelim.capacity();
    ByteBuffer frame;

    if (discardingTooLongFrame) {
        // We've just finished discarding a very large frame.
        // Go back to the initial state.
        discardingTooLongFrame = false;
        buffer.skipBytes(minFrameLength + minDelimLength);
    }
}

```

```

    int tooLongFrameLength = this.tooLongFrameLength;
    this.tooLongFrameLength = 0;
    if (!failFast) {
        fail(ctx, tooLongFrameLength);
    }
    return null;
}

```

```

if (minFrameLength > maxFrameLength) {
    // Discard read frame.
    buffer.skipBytes(minFrameLength + minDelimLength);
    fail(ctx, minFrameLength);
    return null;
}

```

```

if (stripDelimiter) {
    frame = buffer.readBytes(minFrameLength);
    buffer.skipBytes(minDelimLength);
} else {

```

```

        frame = buffer.readBytes(minFrameLength + minDelimLength);
    }

    return frame;
} else {
    if (!discardingTooLongFrame) {
        if (buffer.readableBytes() > maxFrameLength) {
            // Discard the content of the buffer until a delimiter is found.
            tooLongFrameLength = buffer.readableBytes();
            buffer.skipBytes(buffer.readableBytes());
            discardingTooLongFrame = true;
            if (failFast) {
                fail(ctx, tooLongFrameLength);
            }
        }
    } else {
        // Still discarding the buffer since a delimiter is not found.
        tooLongFrameLength += buffer.readableBytes();
        buffer.skipBytes(buffer.readableBytes());
    }
    return null;
}
}

private void fail(ChannelHandlerContext ctx, long frameLength) {
    if (frameLength > 0) {
        ctx.fireExceptionCaught(
            new TooLongFrameException(
                "frame length exceeds " + maxFrameLength +
                ": " + frameLength + " - discarded"));
    } else {
        ctx.fireExceptionCaught(
            new TooLongFrameException(
                "frame length exceeds " + maxFrameLength +
                " - discarding"));
    }
}

/**
 * Returns the number of bytes between the readerIndex of the haystack and
 * the first needle found in the haystack. -1 is returned if no needle is
 * found in the haystack.
 */
private static int indexOf(ByteBuf haystack, ByteBuf needle) {
    for (int i = haystack.readerIndex(); i < haystack.writerIndex(); i++) {
        int haystackIndex = i;
        int needleIndex;
        for (needleIndex = 0; needleIndex < needle.capacity(); needleIndex++) {

```

```

        if (haystack.getBytes(haystackIndex) != needle.getBytes(needleIndex)) {
            break;
        } else {
            haystackIndex++;
            if (haystackIndex == haystack.writerIndex() &&
                needleIndex != needle.capacity() - 1) {
                return -1;
            }
        }
    }

    if (needleIndex == needle.capacity()) {
        // Found the needle from the haystack!
        return i - haystack.readerIndex();
    }
}
return -1;
}

private static void validateDelimiter(ByteBuf delimiter) {
    if (delimiter == null) {
        throw new NullPointerException("delimiter");
    }
    if (!delimiter.isReadable()) {
        throw new IllegalArgumentException("empty delimiter");
    }
}

private static void validateMaxFrameLength(int maxFrameLength) {
    if (maxFrameLength <= 0) {
        throw new IllegalArgumentException(
            "maxFrameLength must be a positive integer: " +
            maxFrameLength);
    }
}
}
/*
 * Copyright 2013 The Netty Project
 *
 * The Netty Project licenses this file to you under the Apache License,
 * version 2.0 (the "License"); you may not use this file except in compliance
 * with the License. You may obtain a copy of the License at:
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the

```

```

* License for the specific language governing permissions and limitations
* under the License.
*/
package io.netty.handler.ssl;

/**
 * Event that is fired once the SSL handshake is complete, which may be because it was successful or there
 * was an error.
 */
public final class SslHandshakeCompletionEvent {

    public static final SslHandshakeCompletionEvent SUCCESS = new SslHandshakeCompletionEvent();

    private final Throwable cause;

    /**
     * Creates a new event that indicates a successful handshake.
     */
    private SslHandshakeCompletionEvent() {
        cause = null;
    }

    /**
     * Creates a new event that indicates an unsuccessful handshake.
     * Use {@link #SUCCESS} to indicate a successful handshake.
     */
    public SslHandshakeCompletionEvent(Throwable cause) {
        if (cause == null) {
            throw new NullPointerException("cause");
        }
        this.cause = cause;
    }

    /**
     * Return {@code true} if the handshake was successful
     */
    public boolean isSuccess() {
        return cause == null;
    }

    /**
     * Return the {@link Throwable} if {@link #isSuccess()} returns {@code false}
     * and so the handshake failed.
     */
    public Throwable cause() {
        return cause;
    }
}

```

}

## 1.65 nm-mercury-data-objects 4.2.0-1

### 1.65.1 Available under license :

NDS Proprietary

Copyright (c) 2010: NDS Limited

PROPRIETARY & CONFIDENTIAL

The copyright of this code and related documentation together with any other associated intellectual property rights are vested in NDS Limited and may not be used except in accordance with the terms of the licence that you have entered into with NDS Limited. Use of this material without an express licence from NDS Limited shall be an infringement of copyright and any other intellectual property rights that may be incorporated with this material.

## 1.66 oasm-mercury-data-objects 1.9.0-2

### 1.66.1 Available under license :

NDS Proprietary

Copyright (c) 2013: NDS Limited

PROPRIETARY & CONFIDENTIAL

The copyright of this code and related documentation together with any other associated intellectual property rights are vested in NDS Limited and may not be used except in accordance with the terms of the licence that you have entered into with NDS Limited. Use of this material without an express licence from NDS Limited shall be an infringement of copyright and any other intellectual property rights that may be incorporated with this material.

## 1.67 offercacher-application 3.58.0-2

### 1.67.1 Available under license :

NDS Proprietary

Copyright (c) 2013: NDS Limited

PROPRIETARY & CONFIDENTIAL

The copyright of this code and related documentation together with any other associated intellectual property rights are vested in NDS Limited and may not be used except in accordance with the terms of the licence that you have entered into with NDS Limited. Use of this material without an express licence from NDS Limited shall be an infringement of copyright and any other intellectual property rights that may be incorporated with this material.

## 1.68 quartz 1.7.3

## 1.68.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product



names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

```
/*
 * Copyright 2001-2009 Terracotta, Inc.
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you may not
 * use this file except in compliance with the License. You may obtain a copy
 * of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
 * License for the specific language governing permissions and limitations
 * under the License.
 */
```

## 1.69 scala-actors 2.10.2

### 1.69.1 Available under license :

Copyright (c) 2002-2014 EPFL  
Copyright (c) 2011-2014 Typesafe, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the EPFL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN

IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
/*      _      *\  
** _____ // ____ Scala API      **  
** /_/ _// _|// / _| (c) 2005-2013, LAMP/EPFL      **  
** _\ \// _/ _|// _/ _| http://scala-lang.org/      **  
** /____^____// |/_/_// ||      **  
**          |          **  
\*          */  
/*
```

\* Written by Dawid Kurzyniec, based on public domain code written by Doug Lea  
\* and publicly available documentation, and released to the public domain, as  
\* explained at <http://creativecommons.org/licenses/publicdomain>  
\*/

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

## 1.70 scala-library 2.10.2

### 1.70.1 Available under license :

SCALA LICENSE

Copyright (c) 2002-2012 EPFL, Lausanne, unless otherwise specified.  
All rights reserved.

This software was developed by the Programming Methods Laboratory of the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

Permission to use, copy, modify, and distribute this software in source or binary form for any purpose with or without fee is hereby granted, provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the EPFL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/\*! jQuery UI - v1.9.0 - 2012-10-05

\* <http://jqueryui.com>

\* Includes: jquery.ui.core.js, jquery.ui.widget.js, jquery.ui.mouse.js, jquery.ui.position.js, jquery.ui.accordion.js, jquery.ui.autocomplete.js, jquery.ui.button.js, jquery.ui.datepicker.js, jquery.ui.dialog.js, jquery.ui.draggable.js, jquery.ui.droppable.js, jquery.ui.effect.js, jquery.ui.effect-blind.js, jquery.ui.effect-bounce.js, jquery.ui.effect-clip.js, jquery.ui.effect-drop.js, jquery.ui.effect-explode.js, jquery.ui.effect-fade.js, jquery.ui.effect-fold.js, jquery.ui.effect-highlight.js, jquery.ui.effect-pulsate.js, jquery.ui.effect-scale.js, jquery.ui.effect-shake.js, jquery.ui.effect-slide.js, jquery.ui.effect-transfer.js, jquery.ui.menu.js, jquery.ui.progressbar.js, jquery.ui.resizable.js, jquery.ui.selectable.js, jquery.ui.slider.js, jquery.ui.sortable.js, jquery.ui.spinner.js, jquery.ui.tabs.js, jquery.ui.tooltip.js

\* Copyright (c) 2012 jQuery Foundation and other contributors Licensed MIT \*/

/\*! jQuery v1.8.2 jquery.com | jquery.org/license \*/

/\*\*

\* @preserve jquery.layout 1.3.0 - Release Candidate 30.62

\* \$Date: 2012-08-04 08:00:00 (Thu, 23 Aug 2012) \$

\* \$Rev: 303006 \$

\*

\* Copyright (c) 2012

\* Fabrizio Balliano (<http://www.fabrizioballiano.net>)

\* Kevin Dalman (<http://allpro.net>)

\*

\* Dual licensed under the GPL (<http://www.gnu.org/licenses/gpl.html>)

\* and MIT (<http://www.opensource.org/licenses/mit-license.php>) licenses.

\*

\* Changelog: <http://layout.jquery-dev.net/changelog.cfm#1.3.0.rc30.62>

\* NOTE: This is a short-term release to patch a couple of bugs.

```

* These bugs are listed as officially fixed in RC30.7, which will be released shortly.
*
* Docs: http://layout.jquery-dev.net/documentation.html
* Tips: http://layout.jquery-dev.net/tips.html
* Help: http://groups.google.com/group/jquery-ui-layout
*/
/**
* jquery.layout.state 1.0
* $Date: 2011-07-16 08:00:00 (Sat, 16 July 2011) $
*
* Copyright (c) 2010
* Kevin Dalman (http://allpro.net)
*
* Dual licensed under the GPL (http://www.gnu.org/licenses/gpl.html)
* and MIT (http://www.opensource.org/licenses/mit-license.php) licenses.
*
* @dependancies: UI Layout 1.3.0.rc30.1 or higher
* @dependancies: $.ui.cookie (above)
*
* @support: http://groups.google.com/group/jquery-ui-layout
*/
/**
* jquery.layout.buttons 1.0
* $Date: 2011-07-16 08:00:00 (Sat, 16 July 2011) $
*
* Copyright (c) 2010
* Kevin Dalman (http://allpro.net)
*
* Dual licensed under the GPL (http://www.gnu.org/licenses/gpl.html)
* and MIT (http://www.opensource.org/licenses/mit-license.php) licenses.
*
* @dependancies: UI Layout 1.3.0.rc30.1 or higher
*
* @support: http://groups.google.com/group/jquery-ui-layout
*
* Docs: [ to come ]
* Tips: [ to come ]
*/
/**
* jquery.layout.browserZoom 1.0
* $Date: 2011-12-29 08:00:00 (Thu, 29 Dec 2011) $
*
* Copyright (c) 2012
* Kevin Dalman (http://allpro.net)
*
* Dual licensed under the GPL (http://www.gnu.org/licenses/gpl.html)
* and MIT (http://www.opensource.org/licenses/mit-license.php) licenses.
*

```

```

* @dependancies: UI Layout 1.3.0.rc30.1 or higher
*
* @support: http://groups.google.com/group/jquery-ui-layout
*
* @todo: Extend logic to handle other problematic zooming in browsers
* @todo: Add hotkey/mousewheel bindings to _instantly_ respond to these zoom event
*/
// +-----+ \\
// and; Raphaël 2.1.0 - JavaScript Vector Library          and; \\
// +-----and; \\
// and; Copyright © 2008-2012 Dmitry Baranovskiy (http://raphaeljs.com) and; \\
// and; Copyright © 2008-2012 Sencha Labs (http://sencha.com)          and; \\
// +-----and; \\
// and; Licensed under the MIT (http://raphaeljs.com/license.html) license.and; \\
// +-----+ \\
/*
* tools.tooltip 1.1.3 - Tooltips done right.
*
* Copyright (c) 2009 Tero Piirainen
* http://flowplayer.org/tools/tooltip.html
*
* Dual licensed under MIT and GPL 2+ licenses
* http://www.opensource.org/licenses
*
* Launch : November 2008
* Date: ${date}
* Revision: ${revision}
*/
/*
* Written by Dawid Kurzyniec, based on public domain code written by Doug Lea
* and publicly available documentation, and released to the public domain, as
* explained at http://creativecommons.org/licenses/publicdomain
*/
// -----
//
// Scalax - The Scala Community Library
// Copyright (c) 2005-8 The Scalax Project. All rights reserved.
//
// The primary distribution site is http://scalax.scalaforge.org/
//
// This software is released under the terms of the Revised BSD License.
// There is NO WARRANTY. See the file LICENSE for the full text.
//
// -----

```

# 1.71 scala-reflect 2.10.2

## 1.71.1 Available under license :

Copyright (c) 2002-2014 EPFL

Copyright (c) 2011-2014 Typesafe, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the EPFL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.72 Simple Logging Facade for Java (SLF4J)

## 1.7.2

### 1.72.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was



submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or

consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and  
limitations under the License.

Copyright (c) 2004-2008 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining  
a copy of this software and associated documentation files (the  
"Software"), to deal in the Software without restriction, including  
without limitation the rights to use, copy, modify, merge, publish,  
distribute, sublicense, and/or sell copies of the Software, and to  
permit persons to whom the Software is furnished to do so, subject to  
the following conditions:

The above copyright notice and this permission notice shall be  
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

/\*

\* Copyright 2001-2004 The Apache Software Foundation.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

/\*\*

\* Copyright (c) 2005 - 2010, James Auldridge

\* All rights reserved.

\*

\* Licensed under the BSD, MIT, and GPL (your choice!) Licenses:

\* <http://code.google.com/p/cookies/wiki/License>

\*

\*/

## 1.73 SLF4J API Module 1.6.1

### 1.73.1 Available under license :

\* Copyright (c) 2004-2007 QOS.ch

\* All rights reserved.

\*

\* Permission is hereby granted, free of charge, to any person obtaining

\* a copy of this software and associated documentation files (the

\* "Software"), to deal in the Software without restriction, including

\* without limitation the rights to use, copy, modify, merge, publish,

\* distribute, sublicense, and/or sell copies of the Software, and to

\* permit persons to whom the Software is furnished to do so, subject to

\* the following conditions:

- \* The above copyright notice and this permission notice shall be
- \* included in all copies or substantial portions of the Software.
- \* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
- \* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
- \* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
- \* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
- \* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
- \* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
- \* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.74 Spring Framework 2.0.7

### 1.74.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.  
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
=====  
== NOTICE file corresponding to section 4 d of the Apache License, ==  
== Version 2.0, in this case for the Spring Framework distribution. ==  
=====
```

This product includes software developed by  
the Apache Software Foundation (<http://www.apache.org>).

The end-user documentation included with a redistribution, if any,  
must include the following acknowledgement:

"This product includes software developed by the Spring Framework  
Project (<http://www.springframework.org>)."

Alternatively, this acknowledgement may appear in the software itself,  
if and wherever such third-party acknowledgements normally appear.

The names "Spring" and "Spring Framework" must not be used to  
endorse or promote products derived from this software without  
prior written permission. For written permission, please contact  
[rod.johnson@interface21.com](mailto:rod.johnson@interface21.com) or [juergen.hoeller@interface21.com](mailto:juergen.hoeller@interface21.com).

/\*

\* Copyright 2002-2007 the original author or authors.



\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

## 1.75 Spring JMS 4.0.0.RELEASE

### 1.75.1 Available under license :

The Apache Software License, Version 2.0  
<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.76 Spring TestContext Framework 4.0.0.RELEASE

### 1.76.1 Available under license :

The Apache Software License, Version 2.0  
<http://www.apache.org/licenses/LICENSE-2.0.txt>

## 1.77 Spring Web 4.0.0.RELEASE

### 1.77.1 Available under license :

/\*  
\* Copyright 2002-2013 the original author or authors.  
\*  
\* Licensed under the Apache License, Version 2.0 (the "License");  
\* you may not use this file except in compliance with the License.  
\* You may obtain a copy of the License at  
\*  
\* <http://www.apache.org/licenses/LICENSE-2.0>  
\*  
\* Unless required by applicable law or agreed to in writing, software  
\* distributed under the License is distributed on an "AS IS" BASIS,  
\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
\* See the License for the specific language governing permissions and  
\* limitations under the License.

\*/

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

## TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously

marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## 1.78 spring-beans 3.2.1.RELEASE

### 1.78.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or

indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.

/\*

\* Copyright 2002-2013 the original author or authors.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

# 1.79 spring-core 3.1.1.RELEASE

## 1.79.1 Available under license :

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic



mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any

additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

### APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

/\*

\* Copyright 2002-2008 the original author or authors.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

## 1.80 spring-core 4.0.0.Release

### 1.80.1 Available under license :

/\*

\* Copyright 2002-2012 the original author or authors.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

```
package org.springframework.util.xml;
```

```
import java.util.ArrayList;
```

```
import java.util.Collections;
```

```
import java.util.HashMap;
```

```
import java.util.Iterator;
```

```
import java.util.List;
```

```
import java.util.Map;
```

```
import javax.xml.XMLConstants;
```

```

import javax.xml.namespace.NamespaceContext;

import org.springframework.util.Assert;

/**
 * Simple {@code javax.xml.namespace.NamespaceContext} implementation. Follows the standard
 * {@code NamespaceContext} contract, and is loadable via a {@code java.util.Map} or
 * {@code java.util.Properties} object
 *
 * @author Arjen Poutsma
 * @since 3.0
 */
public class SimpleNamespaceContext implements NamespaceContext {

    private Map<String, String> prefixToNamespaceUri = new HashMap<String, String>();

    private Map<String, List<String>> namespaceUriToPrefixes = new HashMap<String, List<String>>();

    private String defaultNamespaceUri = "";

    @Override
    public String getNamespaceURI(String prefix) {
        Assert.notNull(prefix, "prefix is null");
        if (XMLConstants.XML_NS_PREFIX.equals(prefix)) {
            return XMLConstants.XML_NS_URI;
        }
        else if (XMLConstants.XMLNS_ATTRIBUTE.equals(prefix)) {
            return XMLConstants.XMLNS_ATTRIBUTE_NS_URI;
        }
        else if (XMLConstants.DEFAULT_NS_PREFIX.equals(prefix)) {
            return defaultNamespaceUri;
        }
        else if (prefixToNamespaceUri.containsKey(prefix)) {
            return prefixToNamespaceUri.get(prefix);
        }
        return "";
    }

    @Override
    public String getPrefix(String namespaceUri) {
        List<String> prefixes = getPrefixesInternal(namespaceUri);
        return prefixes.isEmpty() ? null : (String) prefixes.get(0);
    }

    @Override
    public Iterator<String> getPrefixes(String namespaceUri) {
        return getPrefixesInternal(namespaceUri).iterator();
    }
}

```

```

/**
 * Sets the bindings for this namespace context. The supplied map must consist of string key value pairs.
 *
 * @param bindings the bindings
 */
public void setBindings(Map<String, String> bindings) {
    for (Map.Entry<String, String> entry : bindings.entrySet()) {
        bindNamespaceUri(entry.getKey(), entry.getValue());
    }
}

/**
 * Binds the given namespace as default namespace.
 *
 * @param namespaceUri the namespace uri
 */
public void bindDefaultNamespaceUri(String namespaceUri) {
    bindNamespaceUri(XMLConstants.DEFAULT_NS_PREFIX, namespaceUri);
}

/**
 * Binds the given prefix to the given namespace.
 *
 * @param prefix the namespace prefix
 * @param namespaceUri the namespace uri
 */
public void bindNamespaceUri(String prefix, String namespaceUri) {
    Assert.notNull(prefix, "No prefix given");
    Assert.notNull(namespaceUri, "No namespaceUri given");
    if (XMLConstants.DEFAULT_NS_PREFIX.equals(prefix)) {
        defaultNamespaceUri = namespaceUri;
    }
    else {
        prefixToNamespaceUri.put(prefix, namespaceUri);
        getPrefixesInternal(namespaceUri).add(prefix);
    }
}

/** Removes all declared prefixes. */
public void clear() {
    prefixToNamespaceUri.clear();
}

/**
 * Returns all declared prefixes.
 *
 * @return the declared prefixes

```

```

*/
public Iterator<String> getBoundPrefixes() {
    return prefixToNamespaceUri.keySet().iterator();
}

private List<String> getPrefixesInternal(String namespaceUri) {
    if (defaultNamespaceUri.equals(namespaceUri)) {
        return Collections.singletonList(XMLConstants.DEFAULT_NS_PREFIX);
    }
    else if (XMLConstants.XML_NS_URI.equals(namespaceUri)) {
        return Collections.singletonList(XMLConstants.XML_NS_PREFIX);
    }
    else if (XMLConstants.XMLNS_ATTRIBUTE_NS_URI.equals(namespaceUri)) {
        return Collections.singletonList(XMLConstants.XMLNS_ATTRIBUTE);
    }
    else {
        List<String> list = namespaceUriToPrefixes.get(namespaceUri);
        if (list == null) {
            list = new ArrayList<String>();
            namespaceUriToPrefixes.put(namespaceUri, list);
        }
        return list;
    }
}

/**
 * Removes the given prefix from this context.
 *
 * @param prefix the prefix to be removed
 */
public void removeBinding(String prefix) {
    if (XMLConstants.DEFAULT_NS_PREFIX.equals(prefix)) {
        defaultNamespaceUri = "";
    }
    else {
        String namespaceUri = prefixToNamespaceUri.remove(prefix);
        List<String> prefixes = getPrefixesInternal(namespaceUri);
        prefixes.remove(prefix);
    }
}

/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *

```

```

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```
package org.springframework.util.xml;
```

```
import javax.xml.transform.ErrorListener;
import javax.xml.transform.TransformerException;
```

```
import org.apache.commons.logging.Log;
```

```
/**
 * Simple {@code javax.xml.transform.ErrorListener} implementation:
 * logs warnings using the given Commons Logging logger instance,
 * and rethrows errors to discontinue the XML transformation.
 *
 * @author Juergen Hoeller
 * @since 1.2
 */

```

```
public class SimpleTransformErrorListener implements ErrorListener {
```

```
    private final Log logger;
```

```
    /**
     * Create a new SimpleTransformErrorListener for the given
     * Commons Logging logger instance.
     */

```

```
    public SimpleTransformErrorListener(Log logger) {
        this.logger = logger;
    }

```

```
    @Override
    public void warning(TransformerException ex) throws TransformerException {
        logger.warn("XSLT transformation warning", ex);
    }

```

```
    @Override
    public void error(TransformerException ex) throws TransformerException {
        logger.error("XSLT transformation error", ex);
    }

```

```

@Override
public void fatalError(TransformerException ex) throws TransformerException {
    throw ex;
}

}
/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.util.xml;

import org.apache.commons.logging.Log;
import org.xml.sax.ErrorHandler;
import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;

/**
 * Simple {@code org.xml.sax.ErrorHandler} implementation:
 * logs warnings using the given Commons Logging logger instance,
 * and rethrows errors to discontinue the XML transformation.
 *
 * @author Juergen Hoeller
 * @since 1.2
 */
public class SimpleSaxErrorHandler implements ErrorHandler {

    private final Log logger;

    /**
     * Create a new SimpleSaxErrorHandler for the given
     * Commons Logging logger instance.
     */
    public SimpleSaxErrorHandler(Log logger) {
        this.logger = logger;
    }

```



```

}

@Override
public void warning(SAXParseException ex) throws SAXException {
    logger.warn("Ignored XML validation warning", ex);
}

@Override
public void error(SAXParseException ex) throws SAXException {
    throw ex;
}

@Override
public void fatalError(SAXParseException ex) throws SAXException {
    throw ex;
}

}
/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.core.task;

import java.io.Serializable;
import java.util.concurrent.Callable;
import java.util.concurrent.Future;
import java.util.concurrent.FutureTask;
import java.util.concurrent.ThreadFactory;

import org.springframework.util.Assert;
import org.springframework.util.ConcurrencyThrottleSupport;
import org.springframework.util.CustomizableThreadCreator;
import org.springframework.util.concurrent.ListenableFuture;
import org.springframework.util.concurrent.ListenableFutureTask;

```

```

/**
 * {@link TaskExecutor} implementation that fires up a new Thread for each task,
 * executing it asynchronously.
 *
 * <p>Supports limiting concurrent threads through the "concurrencyLimit"
 * bean property. By default, the number of concurrent threads is unlimited.
 *
 * <p><b>NOTE: This implementation does not reuse threads!</b> Consider a
 * thread-pooling TaskExecutor implementation instead, in particular for
 * executing a large number of short-lived tasks.
 *
 * @author Juergen Hoeller
 * @since 2.0
 * @see #setConcurrencyLimit
 * @see SyncTaskExecutor
 * @see org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor
 * @see org.springframework.scheduling.commonj.WorkManagerTaskExecutor
 */
@SuppressWarnings("serial")
public class SimpleAsyncTaskExecutor extends CustomizableThreadCreator implements
AsyncListenableTaskExecutor, Serializable {

/**
 * Permit any number of concurrent invocations: that is, don't throttle concurrency.
 */
public static final int UNBOUNDED_CONCURRENCY =
ConcurrencyThrottleSupport.UNBOUNDED_CONCURRENCY;

/**
 * Switch concurrency 'off': that is, don't allow any concurrent invocations.
 */
public static final int NO_CONCURRENCY = ConcurrencyThrottleSupport.NO_CONCURRENCY;

/** Internal concurrency throttle used by this executor */
private final ConcurrencyThrottleAdapter concurrencyThrottle = new ConcurrencyThrottleAdapter();

private ThreadFactory threadFactory;

/**
 * Create a new SimpleAsyncTaskExecutor with default thread name prefix.
 */
public SimpleAsyncTaskExecutor() {
super();
}

```

```

/**
 * Create a new SimpleAsyncTaskExecutor with the given thread name prefix.
 * @param threadNamePrefix the prefix to use for the names of newly created threads
 */
public SimpleAsyncTaskExecutor(String threadNamePrefix) {
    super(threadNamePrefix);
}

/**
 * Create a new SimpleAsyncTaskExecutor with the given external thread factory.
 * @param threadFactory the factory to use for creating new Threads
 */
public SimpleAsyncTaskExecutor(ThreadFactory threadFactory) {
    this.threadFactory = threadFactory;
}

/**
 * Specify an external factory to use for creating new Threads,
 * instead of relying on the local properties of this executor.
 * <p>You may specify an inner ThreadFactory bean or also a ThreadFactory reference
 * obtained from JNDI (on a Java EE 6 server) or some other lookup mechanism.
 * @see #setThreadNamePrefix
 * @see #setThreadPriority
 */
public void setThreadFactory(ThreadFactory threadFactory) {
    this.threadFactory = threadFactory;
}

/**
 * Return the external factory to use for creating new Threads, if any.
 */
public final ThreadFactory getThreadFactory() {
    return this.threadFactory;
}

/**
 * Set the maximum number of parallel accesses allowed.
 * -1 indicates no concurrency limit at all.
 * <p>In principle, this limit can be changed at runtime,
 * although it is generally designed as a config time setting.
 * NOTE: Do not switch between -1 and any concrete limit at runtime,
 * as this will lead to inconsistent concurrency counts: A limit
 * of -1 effectively turns off concurrency counting completely.
 * @see #UNBOUNDED_CONCURRENCY
 */
public void setConcurrencyLimit(int concurrencyLimit) {
    this.concurrencyThrottle.setConcurrencyLimit(concurrencyLimit);
}

```

```

}

/**
 * Return the maximum number of parallel accesses allowed.
 */
public final int getConcurrencyLimit() {
    return this.concurrencyThrottle.getConcurrencyLimit();
}

/**
 * Return whether this throttle is currently active.
 * @return { @code true } if the concurrency limit for this instance is active
 * @see #getConcurrencyLimit()
 * @see #setConcurrencyLimit
 */
public final boolean isThrottleActive() {
    return this.concurrencyThrottle.isThrottleActive();
}

/**
 * Executes the given task, within a concurrency throttle
 * if configured (through the superclass's settings).
 * @see #doExecute(Runnable)
 */
@Override
public void execute(Runnable task) {
    execute(task, TIMEOUT_INDEFINITE);
}

/**
 * Executes the given task, within a concurrency throttle
 * if configured (through the superclass's settings).
 * <p>Executes urgent tasks (with 'immediate' timeout) directly,
 * bypassing the concurrency throttle (if active). All other
 * tasks are subject to throttling.
 * @see #TIMEOUT_IMMEDIATE
 * @see #doExecute(Runnable)
 */
@Override
public void execute(Runnable task, long startTimeout) {
    Assert.notNull(task, "Runnable must not be null");
    if (isThrottleActive() && startTimeout > TIMEOUT_IMMEDIATE) {
        this.concurrencyThrottle.beforeAccess();
        doExecute(new ConcurrencyThrottlingRunnable(task));
    }
    else {
        doExecute(task);
    }
}

```

```

    }
}

@Override
public Future<?> submit(Runnable task) {
    FutureTask<Object> future = new FutureTask<Object>(task, null);
    execute(future, TIMEOUT_INDEFINITE);
    return future;
}

@Override
public <T> Future<T> submit(Callable<T> task) {
    FutureTask<T> future = new FutureTask<T>(task);
    execute(future, TIMEOUT_INDEFINITE);
    return future;
}

@Override
public ListenableFuture<?> submitListenable(Runnable task) {
    ListenableFutureTask<Object> future = new ListenableFutureTask<Object>(task, null);
    execute(future, TIMEOUT_INDEFINITE);
    return future;
}

@Override
public <T> ListenableFuture<T> submitListenable(Callable<T> task) {
    ListenableFutureTask<T> future = new ListenableFutureTask<T>(task);
    execute(future, TIMEOUT_INDEFINITE);
    return future;
}

/**
 * Template method for the actual execution of a task.
 * <p>The default implementation creates a new Thread and starts it.
 * @param task the Runnable to execute
 * @see #setThreadFactory
 * @see #createThread
 * @see java.lang.Thread#start()
 */
protected void doExecute(Runnable task) {
    Thread thread = (this.threadFactory != null ? this.threadFactory.newThread(task) : createThread(task));
    thread.start();
}

/**
 * Subclass of the general ConcurrencyThrottleSupport class,
 * making { @code beforeAccess()} and { @code afterAccess()}

```

```

* visible to the surrounding class.
*/
private static class ConcurrencyThrottleAdapter extends ConcurrencyThrottleSupport {

    @Override
    protected void beforeAccess() {
        super.beforeAccess();
    }

    @Override
    protected void afterAccess() {
        super.afterAccess();
    }
}

/**
 * This Runnable calls { @code afterAccess()} after the
 * target Runnable has finished its execution.
 */
private class ConcurrencyThrottlingRunnable implements Runnable {

    private final Runnable target;

    public ConcurrencyThrottlingRunnable(Runnable target) {
        this.target = target;
    }

    @Override
    public void run() {
        try {
            this.target.run();
        }
        finally {
            concurrencyThrottle.afterAccess();
        }
    }
}

/*
 * Copyright 2002-2011 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0

```

- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS,
- \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- \* See the License for the specific language governing permissions and
- \* limitations under the License.

```
package org.springframework.core.env;
```

```
/**
```

```
* Parses a { @code String[] } of command line arguments in order to populate a
* { @link CommandLineArgs } object.
```

```
*
```

```
* <h3>Working with option arguments</h3>
```

```
* Option arguments must adhere to the exact syntax:
```

```
* <pre class="code">--optName[=optValue]</pre>
```

```
* That is, options must be prefixed with "{ @code --}", and may or may not specify a value.
```

```
* If a value is specified, the name and value must be separated <em>without spaces</em>
```

```
* by an equals sign ("=").
```

```
*
```

```
* <h4>Valid examples of option arguments</h4>
```

```
* <pre class="code">
```

```
* --foo
```

```
* --foo=bar
```

```
* --foo="bar then baz"
```

```
* --foo=bar,baz,biz</pre>
```

```
*
```

```
* <h4>Invalid examples of option arguments</h4>
```

```
* <pre class="code">
```

```
* -foo
```

```
* --foo bar
```

```
* --foo = bar
```

```
* --foo=bar --foo=baz --foo=biz</pre>
```

```
*
```

```
* <h3>Working with non-option arguments</h3>
```

```
* Any and all arguments specified at the command line without the "{ @code --}" option
```

```
* prefix will be considered as "non-option arguments" and made available through the
```

```
* { @link CommandLineArgs#getNonOptionArgs() } method.
```

```
*
```

```
* @author Chris Beams
```

```
* @since 3.1
```

```
*/
```

```
class SimpleCommandLineArgsParser {
```

```
/**
```

```
* Parse the given { @code String } array based on the rules described { @linkplain
```

```
* SimpleCommandLineArgsParser above }, returning a fully-populated
```

```

* {@link CommandLineArgs} object.
* @param args command line arguments, typically from a {@code main()} method
*/
public CommandLineArgs parse(String... args) {
    CommandLineArgs commandLineArgs = new CommandLineArgs();
    for (String arg : args) {
        if (arg.startsWith("--")) {
            String optionText = arg.substring(2, arg.length());
            String optionName;
            String optionValue = null;
            if (optionText.contains("=")) {
                optionName = optionText.substring(0, optionText.indexOf("="));
                optionValue = optionText.substring(optionText.indexOf("=")+1, optionText.length());
            }
            else {
                optionName = optionText;
            }
            if (optionName.isEmpty() || (optionValue != null && optionValue.isEmpty())) {
                throw new IllegalArgumentException("Invalid argument syntax: " + arg);
            }
            commandLineArgs.addOptionArg(optionName, optionValue);
        }
        else {
            commandLineArgs.addNonOptionArg(arg);
        }
    }
    return commandLineArgs;
}

}
/*
* Copyright 2002-2013 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package org.springframework.core.env;

```



```
import java.util.List;
```

```
/**
```

```
 * { @link CommandLinePropertySource } implementation backed by a simple String array.
```

```
 *
```

```
 * <h3>Purpose</h3>
```

```
 * This { @code CommandLinePropertySource } implementation aims to provide the simplest
```

```
 * possible approach to parsing command line arguments. As with all { @code
```

```
 * CommandLinePropertySource } implementations, command line arguments are broken into two
```

```
 * distinct groups: <em>option arguments</em> and <em>non-option arguments</em>, as
```

```
 * described below <em>(some sections copied from Javadoc for { @link SimpleCommandLineArgsParser })</em>:
```

```
 *
```

```
 * <h3>Working with option arguments</h3>
```

```
 * Option arguments must adhere to the exact syntax:
```

```
 * <pre class="code">--optName[=optValue]</pre>
```

```
 * That is, options must be prefixed with "{ @code --}", and may or may not specify a value.
```

```
 * If a value is specified, the name and value must be separated <em>without spaces</em>
```

```
 * by an equals sign ("=").
```

```
 *
```

```
 * <h4>Valid examples of option arguments</h4>
```

```
 * <pre class="code">
```

```
 * --foo
```

```
 * --foo=bar
```

```
 * --foo="bar then baz"
```

```
 * --foo=bar,baz,biz</pre>
```

```
 *
```

```
 * <h4>Invalid examples of option arguments</h4>
```

```
 * <pre class="code">
```

```
 * -foo
```

```
 * --foo bar
```

```
 * --foo = bar
```

```
 * --foo=bar --foo=baz --foo=biz</pre>
```

```
 *
```

```
 * <h3>Working with non-option arguments</h3>
```

```
 * Any and all arguments specified at the command line without the "{ @code --}" option
```

```
 * prefix will be considered as "non-option arguments" and made available through the
```

```
 * { @link #getNonOptionArgs() } method.
```

```
 *
```

```
 * <h2>Typical usage</h2>
```

```
 * <pre class="code">
```

```
 * public static void main(String[] args) {
```

```
 *     PropertySource<?> ps = new SimpleCommandLinePropertySource(args);
```

```
 *     // ...
```

```
 * }</pre>
```

```
 *
```

```
 * See { @link CommandLinePropertySource } for complete general usage examples.
```

```
 *
```

```
 * <h3>Beyond the basics</h3>
```

```

*
* <p>When more fully-featured command line parsing is necessary, consider using
* the provided { @link JOptCommandLinePropertySource }, or implement your own
* { @code CommandLinePropertySource } against the command line parsing library of your
* choice!
*
* @author Chris Beams
* @since 3.1
* @see CommandLinePropertySource
* @see JOptCommandLinePropertySource
*/
public class SimpleCommandLinePropertySource extends CommandLinePropertySource<CommandLineArgs> {

    /**
     * Create a new { @code SimpleCommandLinePropertySource } having the default name
     * and backed by the given { @code String[] } of command line arguments.
     * @see CommandLinePropertySource#COMMAND_LINE_PROPERTY_SOURCE_NAME
     * @see CommandLinePropertySource#CommandLinePropertySource(Object)
     */
    public SimpleCommandLinePropertySource(String... args) {
        super(new SimpleCommandLineArgsParser().parse(args));
    }

    /**
     * Create a new { @code SimpleCommandLinePropertySource } having the given name
     * and backed by the given { @code String[] } of command line arguments.
     */
    public SimpleCommandLinePropertySource(String name, String[] args) {
        super(name, new SimpleCommandLineArgsParser().parse(args));
    }

    /**
     * Get the property names for the option arguments.
     */
    @Override
    public String[] getPropertyNames() {
        return source.getOptionNames().toArray(new String[source.getOptionNames().size()]);
    }

    @Override
    protected boolean containsOption(String name) {
        return this.source.containsOption(name);
    }

    @Override
    protected List<String> getOptionValues(String name) {
        return this.source.getOptionValues(name);
    }
}

```

```

@Override
protected List<String> getNonOptionArgs() {
    return this.source.getNonOptionArgs();
}

}
/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.core;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;

import org.springframework.util.Assert;
import org.springframework.util.StringUtils;
import org.springframework.util.StringValueResolver;

/**
 * Simple implementation of the { @link AliasRegistry } interface.
 * Serves as base class for
 * { @link org.springframework.beans.factory.support.BeanDefinitionRegistry }
 * implementations.
 *
 * @author Juergen Hoeller
 * @since 2.5.2
 */
public class SimpleAliasRegistry implements AliasRegistry {

    /** Map from alias to canonical name */
    private final Map<String, String> aliasMap = new ConcurrentHashMap<String, String>(16);

```

```

@Override
public void registerAlias(String name, String alias) {
    Assert.hasText(name, "'name' must not be empty");
    Assert.hasText(alias, "'alias' must not be empty");
    if (alias.equals(name)) {
        this.aliasMap.remove(alias);
    }
    else {
        if (!allowAliasOverriding()) {
            String registeredName = this.aliasMap.get(alias);
            if (registeredName != null && !registeredName.equals(name)) {
                throw new IllegalStateException("Cannot register alias '" + alias + "' for name '" +
                    name + "': It is already registered for name '" + registeredName + "'.");
            }
        }
        checkForAliasCircle(name, alias);
        this.aliasMap.put(alias, name);
    }
}

```

```

/**
 * Return whether alias overriding is allowed.
 * Default is { @code true }.
 */
protected boolean allowAliasOverriding() {
    return true;
}

```

```

@Override
public void removeAlias(String alias) {
    String name = this.aliasMap.remove(alias);
    if (name == null) {
        throw new IllegalStateException("No alias '" + alias + "' registered");
    }
}

```

```

@Override
public boolean isAlias(String name) {
    return this.aliasMap.containsKey(name);
}

```

```

@Override
public String[] getAliases(String name) {
    List<String> result = new ArrayList<String>();
    synchronized (this.aliasMap) {
        retrieveAliases(name, result);
    }
}

```

```

    }
    return StringUtils.toStringArray(result);
}

/**
 * Transitively retrieve all aliases for the given name.
 * @param name the target name to find aliases for
 * @param result the resulting aliases list
 */
private void retrieveAliases(String name, List<String> result) {
    for (Map.Entry<String, String> entry : this.aliasMap.entrySet()) {
        String registeredName = entry.getValue();
        if (registeredName.equals(name)) {
            String alias = entry.getKey();
            result.add(alias);
            retrieveAliases(alias, result);
        }
    }
}

/**
 * Resolve all alias target names and aliases registered in this
 * factory, applying the given StringValueResolver to them.
 * <p>The value resolver may for example resolve placeholders
 * in target bean names and even in alias names.
 * @param valueResolver the StringValueResolver to apply
 */
public void resolveAliases(StringValueResolver valueResolver) {
    Assert.notNull(valueResolver, "StringValueResolver must not be null");
    synchronized (this.aliasMap) {
        Map<String, String> aliasCopy = new HashMap<String, String>(this.aliasMap);
        for (String alias : aliasCopy.keySet()) {
            String registeredName = aliasCopy.get(alias);
            String resolvedAlias = valueResolver.resolveStringValue(alias);
            String resolvedName = valueResolver.resolveStringValue(registeredName);
            if (resolvedAlias.equals(resolvedName)) {
                this.aliasMap.remove(alias);
            }
            else if (!resolvedAlias.equals(alias)) {
                String existingName = this.aliasMap.get(resolvedAlias);
                if (existingName != null && !existingName.equals(resolvedName)) {
                    throw new IllegalStateException(
                        "Cannot register resolved alias '" + resolvedAlias + "' (original: '" + alias +
                        "') for name '" + resolvedName + "': It is already registered for name '" +
                        registeredName + "'");
                }
            }
            checkForAliasCircle(resolvedName, resolvedAlias);
            this.aliasMap.remove(alias);
        }
    }
}

```

```

        this.aliasMap.put(resolvedAlias, resolvedName);
    }
    else if (!registeredName.equals(resolvedName)) {
        this.aliasMap.put(alias, resolvedName);
    }
}
}
}

/**
 * Determine the raw name, resolving aliases to canonical names.
 * @param name the user-specified name
 * @return the transformed name
 */
public String canonicalName(String name) {
    String canonicalName = name;
    // Handle aliasing...
    String resolvedName;
    do {
        resolvedName = this.aliasMap.get(canonicalName);
        if (resolvedName != null) {
            canonicalName = resolvedName;
        }
    }
    while (resolvedName != null);
    return canonicalName;
}

/**
 * Check whether the given name points back to given alias as an alias
 * in the other direction, catching a circular reference upfront and
 * throwing a corresponding IllegalStateException.
 * @param name the candidate name
 * @param alias the candidate alias
 * @see #registerAlias
 */
protected void checkForAliasCircle(String name, String alias) {
    if (alias.equals(canonicalName(name))) {
        throw new IllegalStateException("Cannot register alias " + alias +
            " for name " + name + ": Circular reference - " +
            name + " is a direct or indirect alias for " + alias + " already");
    }
}

}

/**
 * Copyright 2002-2012 the original author or authors.
 *

```

```

* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```
package org.springframework.core.type.classreading;
```

```
import java.io.IOException;
```

```
import org.springframework.core.io.DefaultResourceLoader;
```

```
import org.springframework.core.io.Resource;
```

```
import org.springframework.core.io.ResourceLoader;
```

```
import org.springframework.util.ClassUtils;
```

```
/**
```

```

* Simple implementation of the { @link MetadataReaderFactory } interface,
* creating a new ASM { @link org.springframework.asm.ClassReader } for every request.
*

```

```
* @author Juergen Hoeller
```

```
* @since 2.5
```

```
*/
```

```
public class SimpleMetadataReaderFactory implements MetadataReaderFactory {
```

```
    private final ResourceLoader resourceLoader;
```

```
/**
```

```

* Create a new SimpleMetadataReaderFactory for the default class loader.
*

```

```
*/
```

```
public SimpleMetadataReaderFactory() {
```

```
    this.resourceLoader = new DefaultResourceLoader();
```

```
}
```

```
/**
```

```

* Create a new SimpleMetadataReaderFactory for the given resource loader.
*

```

```
* @param resourceLoader the Spring ResourceLoader to use
```

```
* (also determines the ClassLoader to use)
```

```
*/
```

```
public SimpleMetadataReaderFactory(ResourceLoader resourceLoader) {
```

```
    this.resourceLoader = (resourceLoader != null ? resourceLoader : new DefaultResourceLoader());
```

```

}

/**
 * Create a new SimpleMetadataReaderFactory for the given class loader.
 * @param classLoader the ClassLoader to use
 */
public SimpleMetadataReaderFactory(ClassLoader classLoader) {
    this.resourceLoader =
        (classLoader != null ? new DefaultResourceLoader(classLoader) : new DefaultResourceLoader());
}

/**
 * Return the ResourceLoader that this MetadataReaderFactory has been
 * constructed with.
 */
public final ResourceLoader getResourceLoader() {
    return this.resourceLoader;
}

@Override
public MetadataReader getMetadataReader(String className) throws IOException {
    String resourcePath = ResourceLoader.CLASSPATH_URL_PREFIX +
        ClassUtils.convertClassNameToResourcePath(className) + ClassUtils.CLASS_FILE_SUFFIX;
    return getMetadataReader(this.resourceLoader.getResource(resourcePath));
}

@Override
public MetadataReader getMetadataReader(Resource resource) throws IOException {
    return new SimpleMetadataReader(resource, this.resourceLoader.getClassLoader());
}
}

/**
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.

```



```

*/

package org.springframework.core.type.classreading;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;

import org.springframework.asm.ClassReader;
import org.springframework.core.NestedIOException;
import org.springframework.core.io.Resource;
import org.springframework.core.type.AnnotationMetadata;
import org.springframework.core.type.ClassMetadata;

/**
 * { @link MetadataReader } implementation based on an ASM
 * { @link org.springframework.asm.ClassReader }.
 *
 * <p>Package-visible in order to allow for repackaging the ASM library
 * without effect on users of the { @code core.type } package.
 *
 * @author Juergen Hoeller
 * @author Costin Leau
 * @since 2.5
 */
final class SimpleMetadataReader implements MetadataReader {

    private final Resource resource;

    private final ClassMetadata classMetadata;

    private final AnnotationMetadata annotationMetadata;

    SimpleMetadataReader(Resource resource, ClassLoader classLoader) throws IOException {
        InputStream is = new BufferedInputStream(resource.getInputStream());
        ClassReader classReader;
        try {
            classReader = new ClassReader(is);
        }
        catch (IllegalArgumentException ex) {
            throw new NestedIOException("ASM ClassReader failed to parse class file - " +
                "probably due to a new Java class file version that isn't supported yet: " + resource, ex);
        }
        finally {
            is.close();
        }
    }
}

```

```

AnnotationMetadataReadingVisitor visitor = new AnnotationMetadataReadingVisitor(classLoader);
classReader.accept(visitor, ClassReader.SKIP_DEBUG);

this.annotationMetadata = visitor;
// (since AnnotationMetadataReadingVisitor extends ClassMetadataReadingVisitor)
this.classMetadata = visitor;
this.resource = resource;
}

@Override
public Resource getResource() {
    return this.resource;
}

@Override
public ClassMetadata getClassMetadata() {
    return this.classMetadata;
}

@Override
public AnnotationMetadata getAnnotationMetadata() {
    return this.annotationMetadata;
}
}

```

# 1.81 spring-test 4.1.0

## 1.81.1 Available under license :

Apache License  
 Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
  
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
  
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or,

within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

/\*

\* Copyright 2002-2012 the original author or authors.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* <http://www.apache.org/licenses/LICENSE-2.0>

```
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

## 1.82 spring-web 4.0.0.Release

### 1.82.1 Available under license :

```
/*
* Copyright 2002-2012 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package org.springframework.web.context;
```

```
import javax.servlet.ServletContext;
```

```
import org.springframework.context.ApplicationContext;
```

```
/**
* Interface to provide configuration for a web application. This is read-only while
* the application is running, but may be reloaded if the implementation supports this.
*
* <p>This interface adds a { @code getServletContext() } method to the generic
* ApplicationContext interface, and defines a well-known application attribute name
* that the root context must be bound to in the bootstrap process.
*
* <p>Like generic application contexts, web application contexts are hierarchical.
* There is a single root context per application, while each servlet in the application
* (including a dispatcher servlet in the MVC framework) has its own child context.
*
* <p>In addition to standard application context lifecycle capabilities,
* WebApplicationContext implementations need to detect { @link ServletContextAware }
```

```

* beans and invoke the {@code setServletContext} method accordingly.
*
* @author Rod Johnson
* @author Juergen Hoeller
* @since January 19, 2001
* @see ServletContextAware#setServletContext
*/
public interface WebApplicationContext extends ApplicationContext {

/**
 * Context attribute to bind root WebApplicationContext to on successful startup.
 * <p>Note: If the startup of the root context fails, this attribute can contain
 * an exception or error as value. Use WebApplicationContextUtils for convenient
 * lookup of the root WebApplicationContext.
 * @see org.springframework.web.context.support.WebApplicationContextUtils#getWebApplicationContext
 * @see
 org.springframework.web.context.support.WebApplicationContextUtils#getRequiredWebApplicationContext
 */
String ROOT_WEB_APPLICATION_CONTEXT_ATTRIBUTE = WebApplicationContext.class.getName() +
".ROOT";

/**
 * Scope identifier for request scope: "request".
 * Supported in addition to the standard scopes "singleton" and "prototype".
 */
String SCOPE_REQUEST = "request";

/**
 * Scope identifier for session scope: "session".
 * Supported in addition to the standard scopes "singleton" and "prototype".
 */
String SCOPE_SESSION = "session";

/**
 * Scope identifier for global session scope: "globalSession".
 * Supported in addition to the standard scopes "singleton" and "prototype".
 */
String SCOPE_GLOBAL_SESSION = "globalSession";

/**
 * Scope identifier for the global web application scope: "application".
 * Supported in addition to the standard scopes "singleton" and "prototype".
 */
String SCOPE_APPLICATION = "application";

/**
 * Name of the ServletContext environment bean in the factory.
 * @see javax.servlet.ServletContext

```



```

*/
String SERVLET_CONTEXT_BEAN_NAME = "servletContext";

/**
 * Name of the ServletContext/PortletContext init-params environment bean in the factory.
 * <p>Note: Possibly merged with ServletConfig/PortletConfig parameters.
 * ServletConfig parameters override ServletContext parameters of the same name.
 * @see javax.servlet.ServletContext#getInitParameterNames()
 * @see javax.servlet.ServletContext#getInitParameter(String)
 * @see javax.servlet.ServletConfig#getInitParameterNames()
 * @see javax.servlet.ServletConfig#getInitParameter(String)
 */
String CONTEXT_PARAMETERS_BEAN_NAME = "contextParameters";

/**
 * Name of the ServletContext/PortletContext attributes environment bean in the factory.
 * @see javax.servlet.ServletContext#getAttributeNames()
 * @see javax.servlet.ServletContext#getAttribute(String)
 */
String CONTEXT_ATTRIBUTES_BEAN_NAME = "contextAttributes";

/**
 * Return the standard Servlet API ServletContext for this application.
 * <p>Also available for a Portlet application, in addition to the PortletContext.
 */
ServletContext getServletContext();

}
/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.web.context.support;

import java.io.File;

```

```

import javax.servlet.ServletContext;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ApplicationObjectSupport;
import org.springframework.web.context.ServletContextAware;
import org.springframework.web.context.WebApplicationContext;
import org.springframework.web.util.WebUtils;

/**
 * Convenient superclass for application objects running in a WebApplicationContext.
 * Provides {@code getWebApplicationContext()}, {@code getServletContext()},
 * and {@code getTempDir()} methods.
 *
 * @author Juergen Hoeller
 * @since 28.08.2003
 * @see SpringBeanAutowiringSupport
 */
public abstract class WebApplicationObjectSupport extends ApplicationObjectSupport
    implements ServletContextAware {

    private ServletContext servletContext;

    @Override
    public final void setServletContext(ServletContext servletContext) {
        if (servletContext != this.servletContext) {
            this.servletContext = servletContext;
            if (servletContext != null) {
                initServletContext(servletContext);
            }
        }
    }

    /**
     * Overrides the base class behavior to enforce running in an ApplicationContext.
     * All accessors will throw IllegalStateException if not running in a context.
     * @see #getApplicationContext()
     * @see #getMessageSourceAccessor()
     * @see #getWebApplicationContext()
     * @see #getServletContext()
     * @see #getTempDir()
     */
    @Override
    protected boolean isContextRequired() {
        return true;
    }

    /**

```

```

* Calls { @link #initServletContext(javax.servlet.ServletContext)} if the
* given ApplicationContext is a { @link WebApplicationContext}.
*/
@Override
protected void initApplicationContext(ApplicationContext context) {
    super.initApplicationContext(context);
    if (this.servletContext == null && context instanceof WebApplicationContext) {
        this.servletContext = ((WebApplicationContext) context).getServletContext();
        if (this.servletContext != null) {
            initServletContext(this.servletContext);
        }
    }
}

/**
 * Subclasses may override this for custom initialization based
 * on the ServletContext that this application object runs in.
 * <p>The default implementation is empty. Called by
 * { @link #initApplicationContext(org.springframework.context.ApplicationContext)}
 * as well as { @link #setServletContext(javax.servlet.ServletContext)}.
 * @param servletContext the ServletContext that this application object runs in
 * (never { @code null})
 */
protected void initServletContext(ServletContext servletContext) {
}

/**
 * Return the current application context as WebApplicationContext.
 * <p><b>NOTE:</b> Only use this if you actually need to access
 * WebApplicationContext-specific functionality. Preferably use
 * { @code getApplicationContext()} or { @code getServletContext()}
 * else, to be able to run in non-WebApplicationContext environments as well.
 * @throws IllegalStateException if not running in a WebApplicationContext
 * @see #getApplicationContext()
 */
protected final WebApplicationContext getWebApplicationContext() throws IllegalStateException {
    ApplicationContext ctx = getApplicationContext();
    if (ctx instanceof WebApplicationContext) {
        return (WebApplicationContext) getApplicationContext();
    }
    else if (isContextRequired()) {
        throw new IllegalStateException("WebApplicationObjectSupport instance [" + this +
            "] does not run in a WebApplicationContext but in: " + ctx);
    }
    else {
        return null;
    }
}

```

```

/**
 * Return the current ServletContext.
 * @throws IllegalStateException if not running within a ServletContext
 */
protected final ServletContext getServletContext() throws IllegalStateException {
    if (this.servletContext != null) {
        return this.servletContext;
    }
    ServletContext servletContext = getWebApplicationContext().getServletContext();
    if (servletContext == null && isContextRequired()) {
        throw new IllegalStateException("WebApplicationObjectSupport instance [" + this +
            "] does not run within a ServletContext. Make sure the object is fully configured!");
    }
    return servletContext;
}

/**
 * Return the temporary directory for the current web application,
 * as provided by the servlet container.
 * @return the File representing the temporary directory
 * @throws IllegalStateException if not running within a ServletContext
 * @see org.springframework.web.util.WebUtils#getTempDir(javax.servlet.ServletContext)
 */
protected final File getTempDir() throws IllegalStateException {
    return WebUtils.getTempDir(getServletContext());
}
}

/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.web.context.support;

import javax.servlet.ServletConfig;

```

```

import javax.servlet.ServletContext;

import org.springframework.beans.factory.config.ConfigurableListableBeanFactory;
import org.springframework.context.support.StaticApplicationContext;
import org.springframework.core.env.ConfigurableEnvironment;
import org.springframework.core.io.Resource;
import org.springframework.core.io.support.ResourcePatternResolver;
import org.springframework.ui.context.Theme;
import org.springframework.ui.context.ThemeSource;
import org.springframework.ui.context.support.UiApplicationContextUtils;
import org.springframework.web.context.ConfigurableWebApplicationContext;
import org.springframework.web.context.ServletConfigAware;
import org.springframework.web.context.ServletContextAware;

/**
 * Static { @link org.springframework.web.context.WebApplicationContext }
 * implementation for testing. Not intended for use in production applications.
 *
 * <p>Implements the { @link org.springframework.web.context.ConfigurableWebApplicationContext }
 * interface to allow for direct replacement of an { @link XmlWebApplicationContext },
 * despite not actually supporting external configuration files.
 *
 * <p>Interprets resource paths as servlet context resources, i.e. as paths beneath
 * the web application root. Absolute paths, e.g. for files outside the web app root,
 * can be accessed via "file:" URLs, as implemented by
 * { @link org.springframework.core.io.DefaultResourceLoader }.
 *
 * <p>In addition to the special beans detected by
 * { @link org.springframework.context.support.AbstractApplicationContext },
 * this class detects a bean of type { @link org.springframework.ui.context.ThemeSource }
 * in the context, under the special bean name "themeSource".
 *
 * @author Rod Johnson
 * @author Juergen Hoeller
 * @see org.springframework.ui.context.ThemeSource
 */
public class StaticWebApplicationContext extends StaticApplicationContext
    implements ConfigurableWebApplicationContext, ThemeSource {

    private ServletContext servletContext;

    private ServletConfig servletConfig;

    private String namespace;

    private ThemeSource themeSource;

```

```

public StaticWebApplicationContext() {
    setDisplayName("Root WebApplicationContext");
}

/**
 * Set the ServletContext that this WebApplicationContext runs in.
 */
@Override
public void setServletContext(ServletContext servletContext) {
    this.servletContext = servletContext;
}

@Override
public ServletContext getServletContext() {
    return this.servletContext;
}

@Override
public void setServletConfig(ServletConfig servletConfig) {
    this.servletConfig = servletConfig;
    if (servletConfig != null && this.servletContext == null) {
        this.servletContext = servletConfig.getServletContext();
    }
}

@Override
public ServletConfig getServletConfig() {
    return this.servletConfig;
}

@Override
public void setNamespace(String namespace) {
    this.namespace = namespace;
    if (namespace != null) {
        setDisplayName("WebApplicationContext for namespace " + namespace + "");
    }
}

@Override
public String getNamespace() {
    return this.namespace;
}

/**
 * The { @link StaticWebApplicationContext } class does not support this method.
 * @throws UnsupportedOperationException <b>always</b>
 */

```

```

@Override
public void setConfigLocation(String configLocation) {
    if (configLocation != null) {
        throw new UnsupportedOperationException("StaticWebApplicationContext does not support config locations");
    }
}

/**
 * The {@link StaticWebApplicationContext} class does not support this method.
 * @throws UnsupportedOperationException <b>always</b>
 */
@Override
public void setConfigLocations(String[] configLocations) {
    if (configLocations != null) {
        throw new UnsupportedOperationException("StaticWebApplicationContext does not support config locations");
    }
}

@Override
public String[] getConfigLocations() {
    return null;
}

/**
 * Register request/session scopes, a {@link ServletContextAwareProcessor}, etc.
 */
@Override
protected void postProcessBeanFactory(ConfigurableListableBeanFactory beanFactory) {
    beanFactory.addBeanPostProcessor(new ServletContextAwareProcessor(this.servletContext, this.servletConfig));
    beanFactory.ignoreDependencyInterface(ServletContextAware.class);
    beanFactory.ignoreDependencyInterface(ServletConfigAware.class);

    WebApplicationContextUtils.registerWebApplicationScopes(beanFactory, this.servletContext);
    WebApplicationContextUtils.registerEnvironmentBeans(beanFactory, this.servletContext, this.servletConfig);
}

/**
 * This implementation supports file paths beneath the root of the ServletContext.
 * @see ServletContextResource
 */
@Override
protected Resource getResourceByPath(String path) {
    return new ServletContextResource(this.servletContext, path);
}

/**
 * This implementation supports pattern matching in unexpanded WARs too.

```

```

* @see ServletContextResourcePatternResolver
*/
@Override
protected ResourcePatternResolver getResourcePatternResolver() {
    return new ServletContextResourcePatternResolver(this);
}

/**
* Create and return a new {@link StandardServletEnvironment}.
*/
@Override
protected ConfigurableEnvironment createEnvironment() {
    return new StandardServletEnvironment();
}

/**
* Initialize the theme capability.
*/
@Override
protected void onRefresh() {
    this.themeSource = UiApplicationContextUtils.initThemeSource(this);
}

@Override
protected void initPropertySources() {
    super.initPropertySources();
    WebApplicationContextUtils.initServletPropertySources(
        this.getEnvironment().getPropertySources(), this.servletContext, this.servletConfig);
}

@Override
public Theme getTheme(String themeName) {
    return this.themeSource.getTheme(themeName);
}

}
/*
* Copyright 2002-2009 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```



\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

```
package org.springframework.web.context.support;
```

```
import java.io.IOException;
```

```
import org.springframework.beans.BeansException;
```

```
import org.springframework.beans.factory.support.DefaultListableBeanFactory;
```

```
import org.springframework.beans.factory.xml.ResourceEntityResolver;
```

```
import org.springframework.beans.factory.xml.XmlBeanDefinitionReader;
```

```
/**
```

```
* { @link org.springframework.web.context.WebApplicationContext } implementation
```

```
* which takes its configuration from XML documents, understood by an
```

```
* { @link org.springframework.beans.factory.xml.XmlBeanDefinitionReader }.
```

```
* This is essentially the equivalent of
```

```
* { @link org.springframework.context.support.AbstractXmlApplicationContext }
```

```
* for a web environment.
```

```
*
```

```
* <p>By default, the configuration will be taken from "/WEB-INF/applicationContext.xml"
```

```
* for the root context, and "/WEB-INF/test-servlet.xml" for a context with the namespace
```

```
* "test-servlet" (like for a DispatcherServlet instance with the servlet-name "test").
```

```
*
```

```
* <p>The config location defaults can be overridden via the "contextConfigLocation"
```

```
* context-param of { @link org.springframework.web.context.ContextLoader } and servlet
```

```
* init-param of { @link org.springframework.web.servlet.FrameworkServlet }. Config locations
```

```
* can either denote concrete files like "/WEB-INF/context.xml" or Ant-style patterns
```

```
* like "/WEB-INF/*-context.xml" (see { @link org.springframework.util.PathMatcher }
```

```
* javadoc for pattern details).
```

```
*
```

```
* <p>Note: In case of multiple config locations, later bean definitions will
```

```
* override ones defined in earlier loaded files. This can be leveraged to
```

```
* deliberately override certain bean definitions via an extra XML file.
```

```
*
```

```
* <p><b>For a WebApplicationContext that reads in a different bean definition format,
```

```
* create an analogous subclass of { @link AbstractRefreshableWebApplicationContext }.</b>
```

```
* Such a context implementation can be specified as "contextClass" context-param
```

```
* for ContextLoader or "contextClass" init-param for FrameworkServlet.
```

```
*
```

```
* @author Rod Johnson
```

```
* @author Juergen Hoeller
```

```
* @see #setNamespace
```

```
* @see #setConfigLocations
```

```
* @see org.springframework.beans.factory.xml.XmlBeanDefinitionReader
```

```
* @see org.springframework.web.context.ContextLoader#initWebApplicationContext
```

```
* @see org.springframework.web.servlet.FrameworkServlet#initWebApplicationContext
```

```

*/
public class XmlWebApplicationContext extends AbstractRefreshableWebApplicationContext {

    /** Default config location for the root context */
    public static final String DEFAULT_CONFIG_LOCATION = "/WEB-INF/applicationContext.xml";

    /** Default prefix for building a config location for a namespace */
    public static final String DEFAULT_CONFIG_LOCATION_PREFIX = "/WEB-INF/";

    /** Default suffix for building a config location for a namespace */
    public static final String DEFAULT_CONFIG_LOCATION_SUFFIX = ".xml";

    /**
     * Loads the bean definitions via an XmlBeanDefinitionReader.
     * @see org.springframework.beans.factory.xml.XmlBeanDefinitionReader
     * @see #initBeanDefinitionReader
     * @see #loadBeanDefinitions
     */
    @Override
    protected void loadBeanDefinitions(DefaultListableBeanFactory beanFactory) throws BeansException, IOException
    {
        // Create a new XmlBeanDefinitionReader for the given BeanFactory.
        XmlBeanDefinitionReader beanDefinitionReader = new XmlBeanDefinitionReader(beanFactory);

        // Configure the bean definition reader with this context's
        // resource loading environment.
        beanDefinitionReader.setEnvironment(this.getEnvironment());
        beanDefinitionReader.setResourceLoader(this);
        beanDefinitionReader.setEntityResolver(new ResourceEntityResolver(this));

        // Allow a subclass to provide custom initialization of the reader,
        // then proceed with actually loading the bean definitions.
        initBeanDefinitionReader(beanDefinitionReader);
        loadBeanDefinitions(beanDefinitionReader);
    }

    /**
     * Initialize the bean definition reader used for loading the bean
     * definitions of this context. Default implementation is empty.
     * <p>Can be overridden in subclasses, e.g. for turning off XML validation
     * or using a different XmlBeanDefinitionParser implementation.
     * @param beanDefinitionReader the bean definition reader used by this context
     * @see org.springframework.beans.factory.xml.XmlBeanDefinitionReader#setValidationMode
     * @see org.springframework.beans.factory.xml.XmlBeanDefinitionReader#setDocumentReaderClass
     */
    protected void initBeanDefinitionReader(XmlBeanDefinitionReader beanDefinitionReader) {
    }
}

```

```

/**
 * Load the bean definitions with the given XmlBeanDefinitionReader.
 * <p>The lifecycle of the bean factory is handled by the refreshBeanFactory method;
 * therefore this method is just supposed to load and/or register bean definitions.
 * <p>Delegates to a ResourcePatternResolver for resolving location patterns
 * into Resource instances.
 * @throws java.io.IOException if the required XML document isn't found
 * @see #refreshBeanFactory
 * @see #getConfigLocations
 * @see #getResources
 * @see #getResourcePatternResolver
 */
protected void loadBeanDefinitions(XmlBeanDefinitionReader reader) throws IOException {
    String[] configLocations = getConfigLocations();
    if (configLocations != null) {
        for (String configLocation : configLocations) {
            reader.loadBeanDefinitions(configLocation);
        }
    }
}

/**
 * The default location for the root context is "/WEB-INF/applicationContext.xml",
 * and "/WEB-INF/test-servlet.xml" for a context with the namespace "test-servlet"
 * (like for a DispatcherServlet instance with the servlet-name "test").
 */
@Override
protected String[] getDefaultConfigLocations() {
    if (getNamespace() != null) {
        return new String[] {DEFAULT_CONFIG_LOCATION_PREFIX + getNamespace() +
DEFAULT_CONFIG_LOCATION_SUFFIX};
    }
    else {
        return new String[] {DEFAULT_CONFIG_LOCATION};
    }
}

/**
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 */

```

- \* Unless required by applicable law or agreed to in writing, software
- \* distributed under the License is distributed on an "AS IS" BASIS,
- \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- \* See the License for the specific language governing permissions and
- \* limitations under the License.
- \*/

```
package org.springframework.web.context.support;
```

```
import java.util.Arrays;
import java.util.LinkedHashSet;
import java.util.Set;
```

```
import org.springframework.beans.factory.support.BeanNameGenerator;
import org.springframework.beans.factory.support.DefaultListableBeanFactory;
import org.springframework.context.annotation.AnnotatedBeanDefinitionReader;
import org.springframework.context.annotation.AnnotationConfigUtils;
import org.springframework.context.annotation.ClassPathBeanDefinitionScanner;
import org.springframework.context.annotation.ScopeMetadataResolver;
import org.springframework.util.Assert;
import org.springframework.util.StringUtils;
import org.springframework.web.context.ContextLoader;
```

```
/**
```

- \* { @link org.springframework.web.context.WebApplicationContext WebApplicationContext }
- \* implementation which accepts annotated classes as input - in particular
- \* { @link org.springframework.context.annotation.Configuration @Configuration }-annotated
- \* classes, but also plain { @link org.springframework.stereotype.Component @Component }
- \* classes and JSR-330 compliant classes using { @code javax.inject } annotations. Allows
- \* for registering classes one by one (specifying class names as config location) as well
- \* as for classpath scanning (specifying base packages as config location).
- \*
- \* <p>This is essentially the equivalent of
- \* { @link org.springframework.context.annotation.AnnotationConfigApplicationContext
- \* AnnotationConfigApplicationContext } for a web environment.
- \*
- \* <p>To make use of this application context, the
- \* { @link plain ContextLoader#CONTEXT\_CLASS\_PARAM "contextClass" } context-param for
- \* ContextLoader and/or "contextClass" init-param for FrameworkServlet must be set to
- \* the fully-qualified name of this class.
- \*
- \* <p>As of Spring 3.1, this class may also be directly instantiated and injected into
- \* Spring's { @code DispatcherServlet } or { @code ContextLoaderListener } when using the
- \* new { @link org.springframework.web.WebApplicationInitializer WebApplicationInitializer }
- \* code-based alternative to { @code web.xml }. See its Javadoc for details and usage examples.
- \*
- \* <p>Unlike { @link XmlWebApplicationContext }, no default configuration class locations
- \* are assumed. Rather, it is a requirement to set the

```

* { @linkplain ContextLoader#CONFIG_LOCATION_PARAM "contextConfigLocation" }
* context-param for { @link ContextLoader } and/or "contextConfigLocation" init-param for
* FrameworkServlet. The param-value may contain both fully-qualified
* class names and base packages to scan for components. See { @link #loadBeanDefinitions }
* for exact details on how these locations are processed.
*
* <p>As an alternative to setting the "contextConfigLocation" parameter, users may
* implement an { @link org.springframework.context.ApplicationContextInitializer
* ApplicationContextInitializer } and set the
* { @linkplain ContextLoader#CONTEXT_INITIALIZER_CLASSES_PARAM "contextInitializerClasses" }
* context-param / init-param. In such cases, users should favor the { @link #refresh() }
* and { @link #scan(String...)} methods over the { @link #setConfigLocation(String) }
* method, which is primarily for use by { @code ContextLoader }.
*
* <p>Note: In case of multiple { @code @Configuration } classes, later { @code @Bean }
* definitions will override ones defined in earlier loaded files. This can be leveraged
* to deliberately override certain bean definitions via an extra Configuration class.
*
* @author Chris Beams
* @author Juergen Hoeller
* @since 3.0
* @see org.springframework.context.annotation.AnnotationConfigApplicationContext
*/
public class AnnotationConfigWebApplicationContext extends AbstractRefreshableWebApplicationContext {

    private BeanNameGenerator beanNameGenerator;

    private ScopeMetadataResolver scopeMetadataResolver;

    private final Set<Class<?>> annotatedClasses = new LinkedHashSet<Class<?>>();

    private final Set<String> basePackages = new LinkedHashSet<String>();

    /**
     * Set a custom { @link BeanNameGenerator } for use with { @link AnnotatedBeanDefinitionReader }
     * and/or { @link ClassPathBeanDefinitionScanner }.
     * <p>Default is { @link org.springframework.context.annotation.AnnotationBeanNameGenerator }.
     * @see AnnotatedBeanDefinitionReader#setBeanNameGenerator
     * @see ClassPathBeanDefinitionScanner#setBeanNameGenerator
     */
    public void setBeanNameGenerator(BeanNameGenerator beanNameGenerator) {
        this.beanNameGenerator = beanNameGenerator;
    }

    /**
     * Return the custom { @link BeanNameGenerator } for use with { @link AnnotatedBeanDefinitionReader }
     * and/or { @link ClassPathBeanDefinitionScanner }, if any.

```

```

*/
protected BeanNameGenerator getBeanNameGenerator() {
    return this.beanNameGenerator;
}

/**
 * Set a custom {@link ScopeMetadataResolver} for use with {@link AnnotatedBeanDefinitionReader}
 * and/or {@link ClassPathBeanDefinitionScanner}.
 * <p>Default is an {@link org.springframework.context.annotation.AnnotationScopeMetadataResolver}.
 * @see AnnotatedBeanDefinitionReader#setScopeMetadataResolver
 * @see ClassPathBeanDefinitionScanner#setScopeMetadataResolver
 */
public void setScopeMetadataResolver(ScopeMetadataResolver scopeMetadataResolver) {
    this.scopeMetadataResolver = scopeMetadataResolver;
}

/**
 * Return the custom {@link ScopeMetadataResolver} for use with {@link AnnotatedBeanDefinitionReader}
 * and/or {@link ClassPathBeanDefinitionScanner}, if any.
 */
protected ScopeMetadataResolver getScopeMetadataResolver() {
    return this.scopeMetadataResolver;
}

/**
 * Register one or more annotated classes to be processed.
 * Note that {@link #refresh()} must be called in order for the context
 * to fully process the new class.
 * <p>Calls to {@code register} are idempotent; adding the same
 * annotated class more than once has no additional effect.
 * @param annotatedClasses one or more annotated classes,
 * e.g. {@link org.springframework.context.annotation.Configuration @Configuration} classes
 * @see #scan(String...)
 * @see #loadBeanDefinitions(DefaultListableBeanFactory)
 * @see #setConfigLocation(String)
 * @see #refresh()
 */
public void register(Class<?>... annotatedClasses) {
    Assert.notEmpty(annotatedClasses, "At least one annotated class must be specified");
    this.annotatedClasses.addAll(Arrays.asList(annotatedClasses));
}

/**
 * Perform a scan within the specified base packages.
 * Note that {@link #refresh()} must be called in order for the context to
 * fully process the new class.
 * @param basePackages the packages to check for annotated classes

```

```

* @see #loadBeanDefinitions(DefaultListableBeanFactory)
* @see #register(Class...)
* @see #setConfigLocation(String)
* @see #refresh()
*/
public void scan(String... basePackages) {
    Assert.notEmpty(basePackages, "At least one base package must be specified");
    this.basePackages.addAll(Arrays.asList(basePackages));
}

/**
* Register a { @link org.springframework.beans.factory.config.BeanDefinition } for
* any classes specified by { @link #register(Class...) } and scan any packages
* specified by { @link #scan(String...) }.
* <p>For any values specified by { @link #setConfigLocation(String) } or
* { @link #setConfigLocations(String[]) }, attempt first to load each location as a
* class, registering a { @code BeanDefinition } if class loading is successful,
* and if class loading fails (i.e. a { @code ClassNotFoundException } is raised),
* assume the value is a package and attempt to scan it for annotated classes.
* <p>Enables the default set of annotation configuration post processors, such that
* { @code @Autowired }, { @code @Required }, and associated annotations can be used.
* <p>Configuration class bean definitions are registered with generated bean
* definition names unless the { @code value } attribute is provided to the stereotype
* annotation.
* @see #register(Class...)
* @see #scan(String...)
* @see #setConfigLocation(String)
* @see #setConfigLocations(String[])
* @see AnnotatedBeanDefinitionReader
* @see ClassPathBeanDefinitionScanner
*/
@Override
protected void loadBeanDefinitions(DefaultListableBeanFactory beanFactory) {
    AnnotatedBeanDefinitionReader reader = new AnnotatedBeanDefinitionReader(beanFactory);
    reader.setEnvironment(this.getEnvironment());

    ClassPathBeanDefinitionScanner scanner = new ClassPathBeanDefinitionScanner(beanFactory);
    scanner.setEnvironment(this.getEnvironment());

    BeanNameGenerator beanNameGenerator = getBeanNameGenerator();
    ScopeMetadataResolver scopeMetadataResolver = getScopeMetadataResolver();
    if (beanNameGenerator != null) {
        reader.setBeanNameGenerator(beanNameGenerator);
        scanner.setBeanNameGenerator(beanNameGenerator);
        beanFactory.registerSingleton(
            AnnotationConfigUtils.CONFIGURATION_BEAN_NAME_GENERATOR, beanNameGenerator);
    }
}

```

```

if (scopeMetadataResolver != null) {
    reader.setScopeMetadataResolver(scopeMetadataResolver);
    scanner.setScopeMetadataResolver(scopeMetadataResolver);
}

if (!this.annotatedClasses.isEmpty()) {
    if (logger.isInfoEnabled()) {
        logger.info("Registering annotated classes: [" +
            StringUtils.collectionToCommaDelimitedString(this.annotatedClasses) + "]);
    }
    reader.register(this.annotatedClasses.toArray(new Class<?>[this.annotatedClasses.size()]));
}

if (!this.basePackages.isEmpty()) {
    if (logger.isInfoEnabled()) {
        logger.info("Scanning base packages: [" +
            StringUtils.collectionToCommaDelimitedString(this.basePackages) + "]);
    }
    scanner.scan(this.basePackages.toArray(new String[this.basePackages.size()]));
}

String[] configLocations = getConfigLocations();
if (configLocations != null) {
    for (String configLocation : configLocations) {
        try {
            Class<?> clazz = getClassLoader().loadClass(configLocation);
            if (logger.isInfoEnabled()) {
                logger.info("Successfully resolved class for [" + configLocation + "]);
            }
            reader.register(clazz);
        }
        catch (ClassNotFoundException ex) {
            if (logger.isDebugEnabled()) {
                logger.debug("Could not load class for config location [" + configLocation +
                    "] - trying package scan. " + ex);
            }
            int count = scanner.scan(configLocation);
            if (logger.isInfoEnabled()) {
                if (count == 0) {
                    logger.info("No annotated classes found for specified class/package [" + configLocation + "]);
                }
                else {
                    logger.info("Found " + count + " annotated classes in package [" + configLocation + "]);
                }
            }
        }
    }
}

```



```

}

}
/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.web.context.support;

import javax.servlet.ServletConfig;
import javax.servlet.ServletContext;

import org.springframework.beans.factory.config.ConfigurableListableBeanFactory;
import org.springframework.beans.factory.support.DefaultListableBeanFactory;
import org.springframework.context.support.GenericApplicationContext;
import org.springframework.core.env.ConfigurableEnvironment;
import org.springframework.core.io.Resource;
import org.springframework.core.io.support.ResourcePatternResolver;
import org.springframework.ui.context.Theme;
import org.springframework.ui.context.ThemeSource;
import org.springframework.ui.context.support.UiApplicationContextUtils;
import org.springframework.util.ObjectUtils;
import org.springframework.util.StringUtils;
import org.springframework.web.context.ConfigurableWebApplicationContext;
import org.springframework.web.context.ConfigurableWebEnvironment;
import org.springframework.web.context.ServletContextAware;

/**
 * Subclass of {@link GenericApplicationContext}, suitable for web environments.
 *
 * <p>Implements the
 * {@link org.springframework.web.context.ConfigurableWebApplicationContext},
 * but is not intended for declarative setup in {@code web.xml}. Instead,
 * it is designed for programmatic setup, for example for building nested contexts or
 * for use within Spring 3.1 {@link org.springframework.web.WebApplicationInitializer}s.
 */

```

\* <p><b>If you intend to implement a `WebApplicationContext` that reads bean definitions from configuration files, consider deriving from `AbstractRefreshableWebApplicationContext`, reading the bean definitions in an implementation of the `{ @code loadBeanDefinitions }` method.</b>

\* <p>Interprets resource paths as servlet context resources, i.e. as paths beneath the web application root. Absolute paths, e.g. for files outside the web app root, can be accessed via "file:" URLs, as implemented by `AbstractApplicationContext`.

\* <p>In addition to the special beans detected by `{ @link org.springframework.context.support.AbstractApplicationContext }`, this class detects a `ThemeSource` bean in the context, with the name "themeSource".

\* @author Juergen Hoeller

\* @author Chris Beams

\* @since 1.2

\*/

```
public class GenericWebApplicationContext extends GenericApplicationContext
    implements ConfigurableWebApplicationContext, ThemeSource {
```

```
    private ServletContext servletContext;
```

```
    private ThemeSource themeSource;
```

```
/**
```

```
 * Create a new GenericWebApplicationContext.
```

```
 * @see #setServletContext
```

```
 * @see #registerBeanDefinition
```

```
 * @see #refresh
```

```
*/
```

```
public GenericWebApplicationContext() {
    super();
}
```

```
/**
```

```
 * Create a new GenericWebApplicationContext for the given ServletContext.
```

```
 * @param servletContext the ServletContext to run in
```

```
 * @see #registerBeanDefinition
```

```
 * @see #refresh
```

```
*/
```

```
public GenericWebApplicationContext(ServletContext servletContext) {
    this.servletContext = servletContext;
}
```

```
/**
```

```
 * Create a new GenericWebApplicationContext with the given DefaultListableBeanFactory.
```

```
 * @param beanFactory the DefaultListableBeanFactory instance to use for this context
```

```

* @see #setServletContext
* @see #registerBeanDefinition
* @see #refresh
*/
public GenericWebApplicationContext(DefaultListableBeanFactory beanFactory) {
    super(beanFactory);
}

/**
 * Create a new GenericWebApplicationContext with the given DefaultListableBeanFactory.
 * @param beanFactory the DefaultListableBeanFactory instance to use for this context
 * @param servletContext the ServletContext to run in
 * @see #registerBeanDefinition
 * @see #refresh
 */
public GenericWebApplicationContext(DefaultListableBeanFactory beanFactory, ServletContext servletContext) {
    super(beanFactory);
    this.servletContext = servletContext;
}

/**
 * Set the ServletContext that this WebApplicationContext runs in.
 */
@Override
public void setServletContext(ServletContext servletContext) {
    this.servletContext = servletContext;
}

@Override
public ServletContext getServletContext() {
    return this.servletContext;
}

@Override
public String getApplicationName() {
    return (this.servletContext != null ? this.servletContext.getContextPath() : "");
}

/**
 * Create and return a new {@link StandardServletEnvironment}.
 */
@Override
protected ConfigurableEnvironment createEnvironment() {
    return new StandardServletEnvironment();
}

/**

```

```

* Register ServletContextAwareProcessor.
* @see ServletContextAwareProcessor
*/
@Override
protected void postProcessBeanFactory(ConfigurableListableBeanFactory beanFactory) {
    beanFactory.addBeanPostProcessor(new ServletContextAwareProcessor(this.servletContext));
    beanFactory.ignoreDependencyInterface(ServletContextAware.class);

    WebApplicationContextUtils.registerWebApplicationScopes(beanFactory, this.servletContext);
    WebApplicationContextUtils.registerEnvironmentBeans(beanFactory, this.servletContext);
}

/**
 * This implementation supports file paths beneath the root of the ServletContext.
 * @see ServletContextResource
 */
@Override
protected Resource getResourceByPath(String path) {
    return new ServletContextResource(this.servletContext, path);
}

/**
 * This implementation supports pattern matching in unexpanded WARs too.
 * @see ServletContextResourcePatternResolver
 */
@Override
protected ResourcePatternResolver getResourcePatternResolver() {
    return new ServletContextResourcePatternResolver(this);
}

/**
 * Initialize the theme capability.
 */
@Override
protected void onRefresh() {
    this.themeSource = UiApplicationContextUtils.initThemeSource(this);
}

/**
 * {@inheritDoc}
 * <p>Replace {@code Servlet}-related property sources.
 */
@Override
protected void initPropertySources() {
    super.initPropertySources();
    ConfigurableEnvironment env = this.getEnvironment();
    if (env instanceof ConfigurableWebEnvironment) {
        ((ConfigurableWebEnvironment)env).initPropertySources(

```

```

        this.servletContext, null);
    }
}

@Override
public Theme getTheme(String themeName) {
    return this.themeSource.getTheme(themeName);
}

// -----
// Pseudo-implementation of ConfigurableWebApplicationContext
// -----

@Override
public void setServletConfig(ServletConfig servletConfig) {
    // no-op
}

@Override
public ServletConfig getServletConfig() {
    throw new UnsupportedOperationException(
        "GenericWebApplicationContext does not support getServletConfig()");
}

@Override
public void setNamespace(String namespace) {
    // no-op
}

@Override
public String getNamespace() {
    throw new UnsupportedOperationException(
        "GenericWebApplicationContext does not support getNamespace()");
}

@Override
public void setConfigLocation(String configLocation) {
    if (StringUtils.hasText(configLocation)) {
        throw new UnsupportedOperationException(
            "GenericWebApplicationContext does not support setConfigLocation(). " +
            "Do you still have an 'contextConfigLocations' init-param set?");
    }
}

@Override
public void setConfigLocations(String[] configLocations) {
    if (!ObjectUtils.isEmpty(configLocations)) {

```

```

    throw new UnsupportedOperationException(
        "GenericWebApplicationContext does not support setConfigLocations(). " +
        "Do you still have an 'contextConfigLocations' init-param set?");
    }
}

@Override
public String[] getConfigLocations() {
    throw new UnsupportedOperationException(
        "GenericWebApplicationContext does not support getConfigLocations()");
}

}
/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.web.context.support;

import javax.servlet.ServletConfig;
import javax.servlet.ServletContext;

import org.springframework.beans.factory.config.ConfigurableListableBeanFactory;
import org.springframework.context.support.AbstractRefreshableConfigApplicationContext;
import org.springframework.core.env.ConfigurableEnvironment;
import org.springframework.core.io.Resource;
import org.springframework.core.io.support.ResourcePatternResolver;
import org.springframework.ui.context.Theme;
import org.springframework.ui.context.ThemeSource;
import org.springframework.ui.context.support.UiApplicationContextUtils;
import org.springframework.web.context.ConfigurableWebApplicationContext;
import org.springframework.web.context.ConfigurableWebEnvironment;
import org.springframework.web.context.ServletConfigAware;
import org.springframework.web.context.ServletContextAware;

/**

```

```

* { @link org.springframework.context.support.AbstractRefreshableApplicationContext }
* subclass which implements the
* { @link org.springframework.web.context.ConfigurableWebApplicationContext }
* interface for web environments. Provides a "configLocations" property,
* to be populated through the ConfigurableWebApplicationContext interface
* on web application startup.
*
* <p>This class is as easy to subclass as AbstractRefreshableApplicationContext:
* All you need to implements is the { @link #loadBeanDefinitions } method;
* see the superclass javadoc for details. Note that implementations are supposed
* to load bean definitions from the files specified by the locations returned
* by the { @link #getConfigLocations } method.
*
* <p>Interprets resource paths as servlet context resources, i.e. as paths beneath
* the web application root. Absolute paths, e.g. for files outside the web app root,
* can be accessed via "file:" URLs, as implemented by
* { @link org.springframework.core.io.DefaultResourceLoader }.
*
* <p>In addition to the special beans detected by
* { @link org.springframework.context.support.AbstractApplicationContext },
* this class detects a bean of type { @link org.springframework.ui.context.ThemeSource }
* in the context, under the special bean name "themeSource".
*
* <p><b>This is the web context to be subclassed for a different bean definition format.</b>
* Such a context implementation can be specified as "contextClass" context-param
* for { @link org.springframework.web.context.ContextLoader } or as "contextClass"
* init-param for { @link org.springframework.web.servlet.FrameworkServlet },
* replacing the default { @link XmlWebApplicationContext }. It will then automatically
* receive the "contextConfigLocation" context-param or init-param, respectively.
*
* <p>Note that WebApplicationContext implementations are generally supposed
* to configure themselves based on the configuration received through the
* { @link ConfigurableWebApplicationContext } interface. In contrast, a standalone
* application context might allow for configuration in custom startup code
* (for example, { @link org.springframework.context.support.GenericApplicationContext }).
*
* @author Juergen Hoeller
* @since 1.1.3
* @see #loadBeanDefinitions
* @see org.springframework.web.context.ConfigurableWebApplicationContext#setConfigLocations
* @see org.springframework.ui.context.ThemeSource
* @see XmlWebApplicationContext
*/
public abstract class AbstractRefreshableWebApplicationContext extends
AbstractRefreshableConfigApplicationContext
implements ConfigurableWebApplicationContext, ThemeSource {

/** Servlet context that this context runs in */

```

```

private ServletContext servletContext;

/** Servlet config that this context runs in, if any */
private ServletConfig servletConfig;

/** Namespace of this context, or { @code null} if root */
private String namespace;

/** the ThemeSource for this ApplicationContext */
private ThemeSource themeSource;

public AbstractRefreshableWebApplicationContext() {
    setDisplayName("Root WebApplicationContext");
}

@Override
public void setServletContext(ServletContext servletContext) {
    this.servletContext = servletContext;
}

@Override
public ServletContext getServletContext() {
    return this.servletContext;
}

@Override
public void setServletConfig(ServletConfig servletConfig) {
    this.servletConfig = servletConfig;
    if (servletConfig != null && this.servletContext == null) {
        setServletContext(servletConfig.getServletContext());
    }
}

@Override
public ServletConfig getServletConfig() {
    return this.servletConfig;
}

@Override
public void setNamespace(String namespace) {
    this.namespace = namespace;
    if (namespace != null) {
        setDisplayName("WebApplicationContext for namespace '" + namespace + "'");
    }
}

```



```

@Override
public String getNamespace() {
    return this.namespace;
}

@Override
public String[] getConfigLocations() {
    return super.getConfigLocations();
}

@Override
public String getApplicationName() {
    return (this.servletContext != null ? this.servletContext.getContextPath() : "");
}

/**
 * Create and return a new {@link StandardServletEnvironment}. Subclasses may override
 * in order to configure the environment or specialize the environment type returned.
 */
@Override
protected ConfigurableEnvironment createEnvironment() {
    return new StandardServletEnvironment();
}

/**
 * Register request/session scopes, a {@link ServletContextAwareProcessor}, etc.
 */
@Override
protected void postProcessBeanFactory(ConfigurableListableBeanFactory beanFactory) {
    beanFactory.addBeanPostProcessor(new ServletContextAwareProcessor(this.servletContext, this.servletConfig));
    beanFactory.ignoreDependencyInterface(ServletContextAware.class);
    beanFactory.ignoreDependencyInterface(ServletConfigAware.class);

    WebApplicationContextUtils.registerWebApplicationScopes(beanFactory, this.servletContext);
    WebApplicationContextUtils.registerEnvironmentBeans(beanFactory, this.servletContext, this.servletConfig);
}

/**
 * This implementation supports file paths beneath the root of the ServletContext.
 * @see ServletContextResource
 */
@Override
protected Resource getResourceByPath(String path) {
    return new ServletContextResource(this.servletContext, path);
}

/**
 * This implementation supports pattern matching in unexpanded WARs too.

```

```

* @see ServletContextResourcePatternResolver
*/
@Override
protected ResourcePatternResolver getResourcePatternResolver() {
    return new ServletContextResourcePatternResolver(this);
}

/**
 * Initialize the theme capability.
 */
@Override
protected void onRefresh() {
    this.themeSource = UiApplicationContextUtils.initThemeSource(this);
}

/**
 * {@inheritDoc}
 * <p>Replace {@code Servlet}-related property sources.
 */
@Override
protected void initPropertySources() {
    super.initPropertySources();
    ConfigurableEnvironment env = this.getEnvironment();
    if (env instanceof ConfigurableWebEnvironment) {
        ((ConfigurableWebEnvironment)env).initPropertySources(
            this.servletContext, this.servletConfig);
    }
}

@Override
public Theme getTheme(String themeName) {
    return this.themeSource.getTheme(themeName);
}

}

/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and

```

\* limitations under the License.

\*/

```
package org.springframework.web.context.support;
```

```
import static
```

```
org.springframework.web.context.support.StandardServletEnvironment.SERVLET_CONFIG_PROPERTY_SOURCE_NAME;
```

```
import static
```

```
org.springframework.web.context.support.StandardServletEnvironment.SERVLET_CONTEXT_PROPERTY_SOURCE_NAME;
```

```
import java.io.Serializable;
```

```
import java.util.Collections;
```

```
import java.util.Enumeration;
```

```
import java.util.HashMap;
```

```
import java.util.Map;
```

```
import javax.faces.context.ExternalContext;
```

```
import javax.faces.context.FacesContext;
```

```
import javax.servlet.ServletConfig;
```

```
import javax.servlet.ServletContext;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpSession;
```

```
import org.springframework.beans.factory.ObjectFactory;
```

```
import org.springframework.beans.factory.config.ConfigurableListableBeanFactory;
```

```
import org.springframework.core.env.MutablePropertySources;
```

```
import org.springframework.core.env.PropertySource.StubPropertySource;
```

```
import org.springframework.util.Assert;
```

```
import org.springframework.util.ClassUtils;
```

```
import org.springframework.web.context.ConfigurableWebApplicationContext;
```

```
import org.springframework.web.context.WebApplicationContext;
```

```
import org.springframework.web.context.request.RequestAttributes;
```

```
import org.springframework.web.context.request.RequestContextHolder;
```

```
import org.springframework.web.context.request.RequestScope;
```

```
import org.springframework.web.context.request.ServletRequestAttributes;
```

```
import org.springframework.web.context.request.ServletWebRequest;
```

```
import org.springframework.web.context.request.SessionScope;
```

```
import org.springframework.web.context.request.WebRequest;
```

```
/**
```

```
* Convenience methods for retrieving the root
```

```
* { @link org.springframework.web.context.WebApplicationContext } for a given
```

```
* { @code ServletContext }. This is e.g. useful for accessing a Spring
```

```
* context from within custom web views or Struts actions.
```

```
*
```

```
* <p>Note that there are more convenient ways of accessing the root context for
```

```

* many web frameworks, either part of Spring or available as external library.
* This helper class is just the most generic way to access the root context.
*
* @author Juergen Hoeller
* @see org.springframework.web.context.ContextLoader
* @see org.springframework.web.servlet.FrameworkServlet
* @see org.springframework.web.servlet.DispatcherServlet
* @see org.springframework.web.jsf.FacesContextUtils
* @see org.springframework.web.jsf.el.SpringBeanFacesELResolver
*/
public abstract class WebApplicationContextUtils {

    private static final boolean jsfPresent =
        ClassUtils.isPresent("javax.faces.context.FacesContext", RequestContextHolder.class.getClassLoader());

    /**
     * Find the root WebApplicationContext for this web application, which is
     * typically loaded via { @link org.springframework.web.context.ContextLoaderListener}.
     * <p>Will rethrow an exception that happened on root context startup,
     * to differentiate between a failed context startup and no context at all.
     * @param sc ServletContext to find the web application context for
     * @return the root WebApplicationContext for this web app
     * @throws IllegalStateException if the root WebApplicationContext could not be found
     * @see
     org.springframework.web.context.WebApplicationContext#ROOT_WEB_APPLICATION_CONTEXT_ATTRIBUTE
     */
    public static WebApplicationContext getRequiredWebApplicationContext(ServletContext sc)
        throws IllegalStateException {

        WebApplicationContext wac = getWebApplicationContext(sc);
        if (wac == null) {
            throw new IllegalStateException("No WebApplicationContext found: no ContextLoaderListener registered?");
        }
        return wac;
    }

    /**
     * Find the root WebApplicationContext for this web application, which is
     * typically loaded via { @link org.springframework.web.context.ContextLoaderListener}.
     * <p>Will rethrow an exception that happened on root context startup,
     * to differentiate between a failed context startup and no context at all.
     * @param sc ServletContext to find the web application context for
     * @return the root WebApplicationContext for this web app, or { @code null } if none
     * @see
     org.springframework.web.context.WebApplicationContext#ROOT_WEB_APPLICATION_CONTEXT_ATTRIBUTE
     */

```

```

*/
public static WebApplicationContext getWebApplicationContext(ServletContext sc) {
    return getWebApplicationContext(sc,
WebApplicationContext.ROOT_WEB_APPLICATION_CONTEXT_ATTRIBUTE);
}

/**
 * Find a custom WebApplicationContext for this web application.
 * @param sc ServletContext to find the web application context for
 * @param attrName the name of the ServletContext attribute to look for
 * @return the desired WebApplicationContext for this web app, or { @code null } if none
 */
public static WebApplicationContext getWebApplicationContext(ServletContext sc, String attrName) {
    Assert.notNull(sc, "ServletContext must not be null");
    Object attr = sc.getAttribute(attrName);
    if (attr == null) {
        return null;
    }
    if (attr instanceof RuntimeException) {
        throw (RuntimeException) attr;
    }
    if (attr instanceof Error) {
        throw (Error) attr;
    }
    if (attr instanceof Exception) {
        throw new IllegalStateException((Exception) attr);
    }
    if (!(attr instanceof WebApplicationContext)) {
        throw new IllegalStateException("Context attribute is not of type WebApplicationContext: " + attr);
    }
    return (WebApplicationContext) attr;
}

/**
 * Register web-specific scopes ("request", "session", "globalSession")
 * with the given BeanFactory, as used by the WebApplicationContext.
 * @param beanFactory the BeanFactory to configure
 */
public static void registerWebApplicationScopes(ConfigurableListableBeanFactory beanFactory) {
    registerWebApplicationScopes(beanFactory, null);
}

/**
 * Register web-specific scopes ("request", "session", "globalSession", "application")
 * with the given BeanFactory, as used by the WebApplicationContext.
 * @param beanFactory the BeanFactory to configure
 * @param sc the ServletContext that we're running within

```

```

*/
public static void registerWebApplicationScopes(ConfigurableListableBeanFactory beanFactory, ServletContext sc)
{
    beanFactory.registerScope(WebApplicationContext.SCOPE_REQUEST, new RequestScope());
    beanFactory.registerScope(WebApplicationContext.SCOPE_SESSION, new SessionScope(false));
    beanFactory.registerScope(WebApplicationContext.SCOPE_GLOBAL_SESSION, new SessionScope(true));
    if (sc != null) {
        ServletContextScope appScope = new ServletContextScope(sc);
        beanFactory.registerScope(WebApplicationContext.SCOPE_APPLICATION, appScope);
        // Register as ServletContext attribute, for ContextCleanupListener to detect it.
        sc.setAttribute(ServletContextScope.class.getName(), appScope);
    }

    beanFactory.registerResolvableDependency(ServletRequest.class, new RequestObjectFactory());
    beanFactory.registerResolvableDependency(HttpSession.class, new SessionObjectFactory());
    beanFactory.registerResolvableDependency(WebRequest.class, new WebRequestObjectFactory());
    if (jsfPresent) {
        FacesDependencyRegistrar.registerFacesDependencies(beanFactory);
    }
}

/**
 * Register web-specific environment beans ("contextParameters", "contextAttributes")
 * with the given BeanFactory, as used by the WebApplicationContext.
 * @param bf the BeanFactory to configure
 * @param sc the ServletContext that we're running within
 */
public static void registerEnvironmentBeans(ConfigurableListableBeanFactory bf, ServletContext sc) {
    registerEnvironmentBeans(bf, sc, null);
}

/**
 * Register web-specific environment beans ("contextParameters", "contextAttributes")
 * with the given BeanFactory, as used by the WebApplicationContext.
 * @param bf the BeanFactory to configure
 * @param sc the ServletContext that we're running within
 * @param config the ServletConfig of the containing Portlet
 */
public static void registerEnvironmentBeans(
    ConfigurableListableBeanFactory bf, ServletContext sc, ServletConfig config) {

    if (sc != null && !bf.containsBean(WebApplicationContext.SERVLET_CONTEXT_BEAN_NAME)) {
        bf.registerSingleton(WebApplicationContext.SERVLET_CONTEXT_BEAN_NAME, sc);
    }

    if (config != null &&
        !bf.containsBean(ConfigurableWebApplicationContext.SERVLET_CONFIG_BEAN_NAME)) {
        bf.registerSingleton(ConfigurableWebApplicationContext.SERVLET_CONFIG_BEAN_NAME, config);
    }
}

```

```

}

if (!bf.containsBean(WebApplicationContext.CONTEXT_PARAMETERS_BEAN_NAME)) {
    Map<String, String> parameterMap = new HashMap<String, String>();
    if (sc != null) {
        Enumeration<?> paramNameEnum = sc.getInitParameterNames();
        while (paramNameEnum.hasMoreElements()) {
            String paramName = (String) paramNameEnum.nextElement();
            parameterMap.put(paramName, sc.getInitParameter(paramName));
        }
    }
    if (config != null) {
        Enumeration<?> paramNameEnum = config.getInitParameterNames();
        while (paramNameEnum.hasMoreElements()) {
            String paramName = (String) paramNameEnum.nextElement();
            parameterMap.put(paramName, config.getInitParameter(paramName));
        }
    }
    bf.registerSingleton(WebApplicationContext.CONTEXT_PARAMETERS_BEAN_NAME,
        Collections.unmodifiableMap(parameterMap));
}

if (!bf.containsBean(WebApplicationContext.CONTEXT_ATTRIBUTES_BEAN_NAME)) {
    Map<String, Object> attributeMap = new HashMap<String, Object>();
    if (sc != null) {
        Enumeration<?> attrNameEnum = sc.getAttributeNames();
        while (attrNameEnum.hasMoreElements()) {
            String attrName = (String) attrNameEnum.nextElement();
            attributeMap.put(attrName, sc.getAttribute(attrName));
        }
    }
    bf.registerSingleton(WebApplicationContext.CONTEXT_ATTRIBUTES_BEAN_NAME,
        Collections.unmodifiableMap(attributeMap));
}

}

/**
 * Convenient variant of { @link #initServletPropertySources(MutablePropertySources,
 * ServletContext, ServletConfig)} that always provides { @code null} for the
 * { @link ServletConfig} parameter.
 * @see #initServletPropertySources(MutablePropertySources, ServletContext, ServletConfig)
 */
public static void initServletPropertySources(
    MutablePropertySources propertySources, ServletContext servletContext) {
    initServletPropertySources(propertySources, servletContext, null);
}

/**

```

```

* Replace {@code Servlet}-based {@link StubPropertySource stub property sources} with
* actual instances populated with the given {@code servletContext} and
* {@code servletConfig} objects.
* <p>This method is idempotent with respect to the fact it may be called any number
* of times but will perform replacement of stub property sources with their
* corresponding actual property sources once and only once.
* @param propertySources the {@link MutablePropertySources} to initialize (must not
* be {@code null})
* @param servletContext the current {@link ServletContext} (ignored if {@code null}
* or if the {@link StandardServletEnvironment#SERVLET_CONTEXT_PROPERTY_SOURCE_NAME
* servlet context property source} has already been initialized)
* @param servletConfig the current {@link ServletConfig} (ignored if {@code null}
* or if the {@link StandardServletEnvironment#SERVLET_CONFIG_PROPERTY_SOURCE_NAME
* servlet config property source} has already been initialized)
* @see org.springframework.core.env.PropertySource.StubPropertySource
* @see org.springframework.core.env.ConfigurableEnvironment#getPropertySources()
*/
public static void initServletPropertySources(
    MutablePropertySources propertySources, ServletContext servletContext, ServletConfig servletConfig) {
    Assert.notNull(propertySources, "propertySources must not be null");
    if(servletContext != null &&
        propertySources.contains(SERVLET_CONTEXT_PROPERTY_SOURCE_NAME) &&
        propertySources.get(SERVLET_CONTEXT_PROPERTY_SOURCE_NAME) instanceof StubPropertySource) {
        propertySources.replace(SERVLET_CONTEXT_PROPERTY_SOURCE_NAME, new
ServletContextPropertySource(SERVLET_CONTEXT_PROPERTY_SOURCE_NAME, servletContext));
    }
    if(servletConfig != null &&
        propertySources.contains(SERVLET_CONFIG_PROPERTY_SOURCE_NAME) &&
        propertySources.get(SERVLET_CONFIG_PROPERTY_SOURCE_NAME) instanceof StubPropertySource) {
        propertySources.replace(SERVLET_CONFIG_PROPERTY_SOURCE_NAME, new
ServletConfigPropertySource(SERVLET_CONFIG_PROPERTY_SOURCE_NAME, servletConfig));
    }
}

/**
* Return the current RequestAttributes instance as ServletRequestAttributes.
* @see RequestContextHolder#currentRequestAttributes()
*/
private static ServletRequestAttributes currentRequestAttributes() {
    RequestAttributes requestAttr = RequestContextHolder.currentRequestAttributes();
    if (!(requestAttr instanceof ServletRequestAttributes)) {
        throw new IllegalStateException("Current request is not a servlet request");
    }
    return (ServletRequestAttributes) requestAttr;
}

/**

```



```

* Factory that exposes the current request object on demand.
*/
@SuppressWarnings("serial")
private static class RequestObjectFactory implements ObjectFactory<ServletRequest>, Serializable {

    @Override
    public ServletRequest getObject() {
        return currentRequestAttributes().getRequest();
    }

    @Override
    public String toString() {
        return "Current HttpServletRequest";
    }
}

/**
* Factory that exposes the current session object on demand.
*/
@SuppressWarnings("serial")
private static class SessionObjectFactory implements ObjectFactory<HttpSession>, Serializable {

    @Override
    public HttpSession getObject() {
        return currentRequestAttributes().getRequest().getSession();
    }

    @Override
    public String toString() {
        return "Current HttpSession";
    }
}

/**
* Factory that exposes the current WebRequest object on demand.
*/
@SuppressWarnings("serial")
private static class WebRequestObjectFactory implements ObjectFactory<WebRequest>, Serializable {

    @Override
    public WebRequest getObject() {
        return new ServletWebRequest(currentRequestAttributes().getRequest());
    }

    @Override
    public String toString() {

```

```

return "Current ServletWebRequest";
}
}

/**
 * Inner class to avoid hard-coded JSF dependency.
 */
private static class FacesDependencyRegistrar {

public static void registerFacesDependencies(ConfigurableListableBeanFactory beanFactory) {
beanFactory.registerResolvableDependency(FacesContext.class, new ObjectFactory<FacesContext>() {
@Override
public FacesContext getObject() {
return FacesContext.getCurrentInstance();
}
@Override
public String toString() {
return "Current JSF FacesContext";
}
});
beanFactory.registerResolvableDependency(ExternalContext.class, new ObjectFactory<ExternalContext>() {
@Override
public ExternalContext getObject() {
return FacesContext.getCurrentInstance().getExternalContext();
}
@Override
public String toString() {
return "Current JSF ExternalContext";
}
});
}
}

/**
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and

```

```

* limitations under the License.
*/

package org.springframework.web.context;

import javax.servlet.ServletConfig;
import javax.servlet.ServletContext;

import org.springframework.context.ConfigurableApplicationContext;

/**
 * Interface to be implemented by configurable web application contexts.
 * Supported by {@link ContextLoader} and
 * {@link org.springframework.web.servlet.FrameworkServlet}.
 *
 * <p>Note: The setters of this interface need to be called before an
 * invocation of the {@link #refresh} method inherited from
 * {@link org.springframework.context.ConfigurableApplicationContext}.
 * They do not cause an initialization of the context on their own.
 *
 * @author Juergen Hoeller
 * @since 05.12.2003
 * @see #refresh
 * @see ContextLoader#createWebApplicationContext
 * @see org.springframework.web.servlet.FrameworkServlet#createWebApplicationContext
 */
public interface ConfigurableWebApplicationContext extends WebApplicationContext,
ConfigurableApplicationContext {

/**
 * Prefix for ApplicationContext ids that refer to context path and/or servlet name.
 */
String APPLICATION_CONTEXT_ID_PREFIX = WebApplicationContext.class.getName() + ":";

/**
 * Name of the ServletConfig environment bean in the factory.
 * @see javax.servlet.ServletConfig
 */
String SERVLET_CONFIG_BEAN_NAME = "servletConfig";

/**
 * Set the ServletContext for this web application context.
 * <p>Does not cause an initialization of the context; refresh needs to be
 * called after the setting of all configuration properties.
 * @see #refresh()
 */
void setServletContext(ServletContext servletContext);

```

```

/**
 * Set the ServletConfig for this web application context.
 * Only called for a WebApplicationContext that belongs to a specific Servlet.
 * @see #refresh()
 */
void setServletConfig(ServletConfig servletConfig);

/**
 * Return the ServletConfig for this web application context, if any.
 */
ServletConfig getServletConfig();

/**
 * Set the namespace for this web application context,
 * to be used for building a default context config location.
 * The root web application context does not have a namespace.
 */
void setNamespace(String namespace);

/**
 * Return the namespace for this web application context, if any.
 */
String getNamespace();

/**
 * Set the config locations for this web application context in init-param style,
 * i.e. with distinct locations separated by commas, semicolons or whitespace.
 * <p>If not set, the implementation is supposed to use a default for the
 * given namespace or the root web application context, as appropriate.
 */
void setConfigLocation(String configLocation);

/**
 * Set the config locations for this web application context.
 * <p>If not set, the implementation is supposed to use a default for the
 * given namespace or the root web application context, as appropriate.
 */
void setConfigLocations(String[] configLocations);

/**
 * Return the config locations for this web application context,
 * or { @code null } if none specified.
 */
String[] getConfigLocations();
}

```

```
/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

```
package org.springframework.web.bind.support;
```

```
/**
 * Simple implementation of the {@link SessionStatus} interface,
 * keeping the {@code complete} flag as an instance variable.
 *
 * @author Juergen Hoeller
 * @since 2.5
 */
```

```
public class SimpleSessionStatus implements SessionStatus {
```

```
    private boolean complete = false;
```

```
    @Override
    public void setComplete() {
        this.complete = true;
    }
}
```

```
    @Override
    public boolean isComplete() {
        return this.complete;
    }
}
```

```
/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

```

* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```
package org.springframework.web.util;
```

```

import java.io.Serializable;
import java.net.URI;
import java.util.Collections;
import java.util.LinkedHashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

```

```
import org.springframework.util.Assert;
```

```

/**
 * Represents a URI template. A URI template is a URI-like String that contains variables enclosed
 * by braces ({@code {} }), which can be expanded to produce an actual URI.
 *
 * <p>See {@link #expand(Map)}, {@link #expand(Object[])}, and {@link #match(String)} for example usages.
 *
 * @author Arjen Poutsma
 * @author Juergen Hoeller
 * @since 3.0
 * @see <a href="http://bitworking.org/projects/URI-Templates/">URI Templates</a>
 */

```

```
@SuppressWarnings("serial")
```

```
public class UriTemplate implements Serializable {
```

```
    /** Captures URI template variable names. */
```

```
    private static final Pattern NAMES_PATTERN = Pattern.compile("\\{([^/]+?)\\}");
```

```
    /** Replaces template variables in the URI template. */
```

```
    private static final String DEFAULT_VARIABLE_PATTERN = "(.*)";
```

```
    private final UriComponents uriComponents;
```

```
    private final List<String> variableNames;
```

```
    private final Pattern matchPattern;
```

```

private final String uriTemplate;

/**
 * Construct a new {@code UriTemplate} with the given URI String.
 * @param uriTemplate the URI template string
 */
public UriTemplate(String uriTemplate) {
    Parser parser = new Parser(uriTemplate);
    this.uriTemplate = uriTemplate;
    this.variableNames = parser.getVariableNames();
    this.matchPattern = parser.getMatchPattern();
    this.uriComponents = UriComponentsBuilder.fromUriString(uriTemplate).build();
}

/**
 * Return the names of the variables in the template, in order.
 * @return the template variable names
 */
public List<String> getVariableNames() {
    return this.variableNames;
}

/**
 * Given the Map of variables, expands this template into a URI. The Map keys represent variable names,
 * the Map values variable values. The order of variables is not significant.
 * <p>Example:
 * <pre class="code">
 * UriTemplate template = new UriTemplate("http://example.com/hotels/{hotel}/bookings/{booking}");
 * Map<String, String> uriVariables = new HashMap<String, String>();
 * uriVariables.put("booking", "42");
 * uriVariables.put("hotel", "1");
 * System.out.println(template.expand(uriVariables));
 * </pre>
 * will print: <blockquote>{@code http://example.com/hotels/1/bookings/42}</blockquote>
 * @param uriVariables the map of URI variables
 * @return the expanded URI
 * @throws IllegalArgumentException if {@code uriVariables} is {@code null};
 * or if it does not contain values for all the variable names
 */
public URI expand(Map<String, ?> uriVariables) {
    UriComponents expandedComponents = uriComponents.expand(uriVariables);
    UriComponents encodedComponents = expandedComponents.encode();
    return encodedComponents.toUri();
}

```

```

/**
 * Given an array of variables, expand this template into a full URI. The array represent variable values.
 * The order of variables is significant.
 * <p>Example:
 * <pre class="code">
 * UriTemplate template = new UriTemplate("http://example.com/hotels/{hotel}/bookings/{booking}");
 * System.out.println(template.expand("1", "42"));
 * </pre>
 * will print: <blockquote>{@code http://example.com/hotels/1/bookings/42}</blockquote>
 * @param uriVariableValues the array of URI variables
 * @return the expanded URI
 * @throws IllegalArgumentException if {@code uriVariables} is {@code null}
 * or if it does not contain sufficient variables
 */
public URI expand(Object... uriVariableValues) {
    UriComponents expandedComponents = uriComponents.expand(uriVariableValues);
    UriComponents encodedComponents = expandedComponents.encode();
    return encodedComponents.toUri();
}

/**
 * Indicate whether the given URI matches this template.
 * @param uri the URI to match to
 * @return {@code true} if it matches; {@code false} otherwise
 */
public boolean matches(String uri) {
    if (uri == null) {
        return false;
    }
    Matcher matcher = this.matchPattern.matcher(uri);
    return matcher.matches();
}

/**
 * Match the given URI to a map of variable values. Keys in the returned map are variable names,
 * values are variable values, as occurred in the given URI.
 * <p>Example:
 * <pre class="code">
 * UriTemplate template = new UriTemplate("http://example.com/hotels/{hotel}/bookings/{booking}");
 * System.out.println(template.match("http://example.com/hotels/1/bookings/42"));
 * </pre>
 * will print: <blockquote>{@code {hotel=1, booking=42}}</blockquote>
 * @param uri the URI to match to
 * @return a map of variable values
 */
public Map<String, String> match(String uri) {
    Assert.notNull(uri, "'uri' must not be null");
    Map<String, String> result = new LinkedHashMap<String, String>(this.variableNames.size());

```



```

Matcher matcher = this.matchPattern.matcher(uri);
if (matcher.find()) {
    for (int i = 1; i <= matcher.groupCount(); i++) {
        String name = this.variableNames.get(i - 1);
        String value = matcher.group(i);
        result.put(name, value);
    }
}
return result;
}

@Override
public String toString() {
    return this.uriTemplate;
}

/**
 * Static inner class to parse URI template strings into a matching regular expression.
 */
private static class Parser {

    private final List<String> variableNames = new LinkedList<String>();

    private final StringBuilder patternBuilder = new StringBuilder();

    private Parser(String uriTemplate) {
        Assert.hasText(uriTemplate, "'uriTemplate' must not be null");
        Matcher m = NAMES_PATTERN.matcher(uriTemplate);
        int end = 0;
        while (m.find()) {
            this.patternBuilder.append(quote(uriTemplate, end, m.start()));
            String match = m.group(1);
            int colonIdx = match.indexOf(':');
            if (colonIdx == -1) {
                this.patternBuilder.append(DEFAULT_VARIABLE_PATTERN);
                this.variableNames.add(match);
            }
            else {
                if (colonIdx + 1 == match.length()) {
                    throw new IllegalArgumentException(
                        "No custom regular expression specified after ':' in \"" + match + "\"");
                }
                String variablePattern = match.substring(colonIdx + 1, match.length());
                this.patternBuilder.append('(');
                this.patternBuilder.append(variablePattern);
                this.patternBuilder.append(')');
                String variableName = match.substring(0, colonIdx);

```

```

    this.variableNames.add(variableName);
    }
    end = m.end();
    }
    this.patternBuilder.append(quote(uriTemplate, end, uriTemplate.length()));
    int lastIdx = this.patternBuilder.length() - 1;
    if (lastIdx >= 0 && this.patternBuilder.charAt(lastIdx) == '/') {
        this.patternBuilder.deleteCharAt(lastIdx);
    }
    }

    private String quote(String fullPath, int start, int end) {
        if (start == end) {
            return "";
        }
        return Pattern.quote(fullPath.substring(start, end));
    }

    private List<String> getVariableNames() {
        return Collections.unmodifiableList(this.variableNames);
    }

    private Pattern getMatchPattern() {
        return Pattern.compile(this.patternBuilder.toString());
    }
    }
}

/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.web.jsf.el;

import java.beans.FeatureDescriptor;
import java.util.Iterator;

```

```

import javax.el.ELContext;
import javax.el.ELException;
import javax.el.ELResolver;
import javax.faces.context.FacesContext;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import org.springframework.beans.BeansException;
import org.springframework.web.context.WebApplicationContext;
import org.springframework.web.jsf.FacesContextUtils;

/**
 * Special JSF { @code ELResolver } that exposes the Spring { @code WebApplicationContext }
 * instance under a variable named "webApplicationContext".
 *
 * <p>In contrast to { @link SpringBeanFacesELResolver }, this ELResolver variant
 * does <i>not</i> resolve JSF variable names as Spring bean names. It rather
 * exposes Spring's root WebApplicationContext <i>itself</i> under a special name,
 * and is able to resolve "webApplicationContext.mySpringManagedBusinessObject"
 * dereferences to Spring-defined beans in that application context.
 *
 * <p>Configure this resolver in your { @code faces-config.xml } file as follows:
 *
 * <pre class="code">
 * <application>
 * ...
 * <el-resolver>org.springframework.web.jsf.el.WebApplicationContextFacesELResolver</el-resolver>
 * </application></pre>
 *
 * @author Juergen Hoeller
 * @since 2.5
 * @see SpringBeanFacesELResolver
 * @see org.springframework.web.jsf.FacesContextUtils#getWebApplicationContext
 */
public class WebApplicationContextFacesELResolver extends ELResolver {

    /**
     * Name of the exposed WebApplicationContext variable: "webApplicationContext".
     */
    public static final String WEB_APPLICATION_CONTEXT_VARIABLE_NAME = "webApplicationContext";

    /** Logger available to subclasses */
    protected final Log logger = LogFactory.getLog(getClass());

    @Override

```

```

public Object getValue(ELContext elContext, Object base, Object property) throws ELException {
    if (base != null) {
        if (base instanceof WebApplicationContext) {
            WebApplicationContext wac = (WebApplicationContext) base;
            String beanName = property.toString();
            if (logger.isTraceEnabled()) {
                logger.trace("Attempting to resolve property '" + beanName + "' in root WebApplicationContext");
            }
            if (wac.containsBean(beanName)) {
                if (logger.isDebugEnabled()) {
                    logger.debug("Successfully resolved property '" + beanName + "' in root WebApplicationContext");
                }
                elContext.setPropertyResolved(true);
                try {
                    return wac.getBean(beanName);
                }
                catch (BeansException ex) {
                    throw new ELException(ex);
                }
            }
            else {
                // Mimic standard JSF/JSP behavior when base is a Map by returning null.
                return null;
            }
        }
        else {
            if (WEB_APPLICATION_CONTEXT_VARIABLE_NAME.equals(property)) {
                elContext.setPropertyResolved(true);
                return getWebApplicationContext(elContext);
            }
        }
    }
    return null;
}

```

@Override

```

public Class<?> getType(ELContext elContext, Object base, Object property) throws ELException {
    if (base != null) {
        if (base instanceof WebApplicationContext) {
            WebApplicationContext wac = (WebApplicationContext) base;
            String beanName = property.toString();
            if (logger.isDebugEnabled()) {
                logger.debug("Attempting to resolve property '" + beanName + "' in root WebApplicationContext");
            }
            if (wac.containsBean(beanName)) {
                if (logger.isDebugEnabled()) {
                    logger.debug("Successfully resolved property '" + beanName + "' in root WebApplicationContext");
                }
            }
        }
    }
    return null;
}

```

```

    }
    elContext.setPropertyResolved(true);
    try {
        return wac.getType(beanName);
    }
    catch (BeansException ex) {
        throw new ELException(ex);
    }
    }
    else {
        // Mimic standard JSF/JSP behavior when base is a Map by returning null.
        return null;
    }
    }
    }
    else {
        if (WEB_APPLICATION_CONTEXT_VARIABLE_NAME.equals(property)) {
            elContext.setPropertyResolved(true);
            return WebApplicationContext.class;
        }
    }

    return null;
}

@Override
public void setValue(ELContext elContext, Object base, Object property, Object value) throws ELException {
}

@Override
public boolean isReadOnly(ELContext elContext, Object base, Object property) throws ELException {
    if (base instanceof WebApplicationContext) {
        elContext.setPropertyResolved(true);
        return false;
    }
    return false;
}

@Override
public Iterator<FeatureDescriptor> getFeatureDescriptors(ELContext elContext, Object base) {
    return null;
}

@Override
public Class<?> getCommonPropertyType(ELContext elContext, Object base) {
    return Object.class;
}

```

```

/**
 * Retrieve the WebApplicationContext reference to expose.
 * <p>The default implementation delegates to FacesContextUtils,
 * returning { @code null } if no WebApplicationContext found.
 * @param elContext the current JSF ELContext
 * @return the Spring web application context
 * @see org.springframework.web.jsf.FacesContextUtils#getWebApplicationContext
 */
protected WebApplicationContext getWebApplicationContext(ELContext elContext) {
    FacesContext facesContext = FacesContext.getCurrentInstance();
    return FacesContextUtils.getRequiredWebApplicationContext(facesContext);
}

}
/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.web;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;

/**
 * Interface to be implemented in Servlet 3.0+ environments in order to configure the
 * { @link ServletContext } programmatically -- as opposed to (or possibly in conjunction
 * with) the traditional { @code web.xml }-based approach.
 *
 * <p>Implementations of this SPI will be detected automatically by { @link
 * SpringServletContainerInitializer }, which itself is bootstrapped automatically
 * by any Servlet 3.0 container. See { @linkplain SpringServletContainerInitializer its
 * Javadoc } for details on this bootstrapping mechanism.
 *
 * <h2>Example</h2>
 * <h3>The traditional, XML-based approach</h3>

```

\* Most Spring users building a web application will need to register Spring's { @code DispatcherServlet }. For reference, in WEB-INF/web.xml, this would typically be done as follows:

```

* <pre class="code">
* { @code
* <servlet>
*   <servlet-name>dispatcher</servlet-name>
*   <servlet-class>
*     org.springframework.web.servlet.DispatcherServlet
*   </servlet-class>
*   <init-param>
*     <param-name>contextConfigLocation</param-name>
*     <param-value>/WEB-INF/spring/dispatcher-config.xml</param-value>
*   </init-param>
*   <load-on-startup>1</load-on-startup>
* </servlet>
*
* <servlet-mapping>
*   <servlet-name>dispatcher</servlet-name>
*   <url-pattern>/</url-pattern>
* </servlet-mapping>}</pre>

```

\* <h3>The code-based approach with { @code WebApplicationInitializer}</h3>

\* Here is the equivalent { @code DispatcherServlet } registration logic, { @code WebApplicationInitializer }-style:

```

* <pre class="code">
* public class MyWebAppInitializer implements WebApplicationInitializer {
*
*   &#064;Override
*   public void onStartUp(ServletContext container) {
*     XmlWebApplicationContext appContext = new XmlWebApplicationContext();
*     appContext.setConfigLocation("/WEB-INF/spring/dispatcher-config.xml");
*
*     ServletRegistration.Dynamic dispatcher =
*       container.addServlet("dispatcher", new DispatcherServlet(appContext));
*     dispatcher.setLoadOnStartup(1);
*     dispatcher.addMapping("/");
*   }
*
* }</pre>

```

\* As an alternative to the above, you can also extend from { @link org.springframework.web.servlet.support.AbstractDispatcherServletInitializer }.

\* As you can see, thanks to Servlet 3.0's new { @link ServletContext#addServlet } method we're actually registering an <em>instance</em> of the { @code DispatcherServlet }, and this means that the { @code DispatcherServlet } can now be treated like any other object -- receiving constructor injection of its application context in this case.

\*  
 \* <p>This style is both simpler and more concise. There is no concern for dealing with  
 \* init-params, etc, just normal JavaBean-style properties and constructor arguments. You  
 \* are free to create and work with your Spring application contexts as necessary before  
 \* injecting them into the { @code DispatcherServlet }.

\*  
 \* <p>Most major Spring Web components have been updated to support this style of  
 \* registration. You'll find that { @code DispatcherServlet }, { @code FrameworkServlet },  
 \* { @code ContextLoaderListener } and { @code DelegatingFilterProxy } all now support  
 \* constructor arguments. Even if a component (e.g. non-Spring, other third party) has not  
 \* been specifically updated for use within { @code WebApplicationInitializers }, they still  
 \* may be used in any case. The Servlet 3.0 { @code ServletContext } API allows for setting  
 \* init-params, context-params, etc programmatically.

\*  
 \* <h2>A 100% code-based approach to configuration</h2>  
 \* In the example above, { @code WEB-INF/web.xml } was successfully replaced with code in  
 \* the form of a { @code WebApplicationInitializer }, but the actual  
 \* { @code dispatcher-config.xml } Spring configuration remained XML-based.  
 \* { @code WebApplicationInitializer } is a perfect fit for use with Spring's code-based  
 \* { @code @Configuration } classes. See @ { @link  
 \* org.springframework.context.annotation.Configuration Configuration } Javadoc for  
 \* complete details, but the following example demonstrates refactoring to use Spring's  
 \* { @link org.springframework.web.context.support.AnnotationConfigWebApplicationContext  
 \* AnnotationConfigWebApplicationContext } in lieu of { @code XmlWebApplicationContext }, and  
 \* user-defined { @code @Configuration } classes { @code AppConfig } and  
 \* { @code DispatcherConfig } instead of Spring XML files. This example also goes a bit  
 \* beyond those above to demonstrate typical configuration of the 'root' application  
 \* context and registration of the { @code ContextLoaderListener }:

```
* <pre class="code">
* public class MyWebAppInitializer implements WebApplicationInitializer {
*
*     &#064;Override
*     public void onStartUp(ServletContext container) {
*         // Create the 'root' Spring application context
*         AnnotationConfigWebApplicationContext rootContext =
*             new AnnotationConfigWebApplicationContext();
*         rootContext.register(AppConfig.class);
*
*         // Manage the lifecycle of the root application context
*         container.addListener(new ContextLoaderListener(rootContext));
*
*         // Create the dispatcher servlet's Spring application context
*         AnnotationConfigWebApplicationContext dispatcherContext =
*             new AnnotationConfigWebApplicationContext();
*         dispatcherContext.register(DispatcherConfig.class);
*
*         // Register and map the dispatcher servlet
*         ServletRegistration.Dynamic dispatcher =
```



```

*   container.addServlet("dispatcher", new DispatcherServlet(dispatcherContext));
*   dispatcher.setLoadOnStartup(1);
*   dispatcher.addMapping("/");
* }
*
* }</pre>
*
* As an alternative to the above, you can also extend from { @link
* org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer}.
*
* Remember that { @code WebApplicationInitializer } implementations are <em>detected
* automatically</em> -- so you are free to package them within your application as you
* see fit.
*
* <h2>Ordering { @code WebApplicationInitializer } execution</h2>
* { @code WebApplicationInitializer } implementations may optionally be annotated at the
* class level with Spring's @ { @link org.springframework.core.annotation.Order Order }
* annotation or may implement Spring's { @link org.springframework.core.Ordered Ordered }
* interface. If so, the initializers will be ordered prior to invocation. This provides
* a mechanism for users to ensure the order in which servlet container initialization
* occurs. Use of this feature is expected to be rare, as typical applications will likely
* centralize all container initialization within a single { @code WebApplicationInitializer }.
*
* <h2>Caveats</h2>
*
* <h3>web.xml versioning</h3>
* <p>{ @code WEB-INF/web.xml } and { @code WebApplicationInitializer } use are not mutually
* exclusive; for example, web.xml can register one servlet, and a { @code
* WebApplicationInitializer } can register another. An initializer can even
* <em>modify</em> registrations performed in { @code web.xml } through methods such as
* { @link ServletContext#getServletRegistration(String) }. <strong>However, if
* { @code WEB-INF/web.xml } is present in the application, its { @code version } attribute
* must be set to "3.0" or greater, otherwise { @code ServletContainerInitializer }
* bootstrapping will be ignored by the servlet container.</strong>
*
* <h3>Mapping to '/' under Tomcat</h3>
* <p>Apache Tomcat maps its internal { @code DefaultServlet } to "/", and on Tomcat versions
* &lt;= 7.0.14, this servlet mapping <em>cannot be overridden programmatically</em>.
* 7.0.15 fixes this issue. Overriding the "/" servlet mapping has also been tested
* successfully under GlassFish 3.1.</p>
*
* @author Chris Beams
* @since 3.1
* @see SpringServletContainerInitializer
* @see org.springframework.web.context.AbstractContextLoaderInitializer
* @see org.springframework.web.servlet.support.AbstractDispatcherServletInitializer
* @see org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer
*/

```

```

public interface WebApplicationInitializer {

    /**
     * Configure the given {@link ServletContext} with any servlets, filters, listeners
     * context-params and attributes necessary for initializing this web application. See
     * examples {@linkplain WebApplicationInitializer above}.
     * @param servletContext the {@code ServletContext} to initialize
     * @throws ServletException if any call against the given {@code ServletContext}
     * throws a {@code ServletException}
     */
    void onStartUp(ServletContext servletContext) throws ServletException;

}

/**
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.web.client;

import java.io.IOException;
import java.lang.reflect.Type;
import java.net.URI;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import java.util.Set;

import javax.xml.transform.Source;

import org.springframework.core.ParameterizedTypeReference;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.http.client.ClientHttpRequest;

```

```

import org.springframework.http.client.ClientHttpRequestFactory;
import org.springframework.http.client.ClientHttpResponse;
import org.springframework.http.client.support.InterceptingHttpAccessor;
import org.springframework.http.converter.ByteArrayHttpMessageConverter;
import org.springframework.http.converter.GenericHttpMessageConverter;
import org.springframework.http.converter.HttpMessageConverter;
import org.springframework.http.converter.ResourceHttpMessageConverter;
import org.springframework.http.converter.StringHttpMessageConverter;
import org.springframework.http.converter.feed.AtomFeedHttpMessageConverter;
import org.springframework.http.converter.feed.RssChannelHttpMessageConverter;
import org.springframework.http.converter.json.MappingJackson2HttpMessageConverter;
import org.springframework.http.converter.support.AllEncompassingFormHttpMessageConverter;
import org.springframework.http.converter.xml.Jaxb2RootElementHttpMessageConverter;
import org.springframework.http.converter.xml.SourceHttpMessageConverter;
import org.springframework.util.Assert;
import org.springframework.util.ClassUtils;
import org.springframework.web.util.UriTemplate;

```

```
/**
```

```
* <strong>Spring's central class for synchronous client-side HTTP access.</strong>
```

```
* It simplifies communication with HTTP servers, and enforces RESTful principles.
```

```
* It handles HTTP connections, leaving application code to provide URLs
```

```
* (with possible template variables) and extract results.
```

```
*
```

```
* <p>The main entry points of this template are the methods named after the six main HTTP methods:
```

```
* <table>
```

```
* <tr><th>HTTP method</th><th>RestTemplate methods</th></tr>
```

```
* <tr><td>DELETE</td><td>{ @link #delete }</td></tr>
```

```
* <tr><td>GET</td><td>{ @link #getForObject }</td></tr>
```

```
* <tr><td></td><td>{ @link #getForEntity }</td></tr>
```

```
* <tr><td>HEAD</td><td>{ @link #headForHeaders }</td></tr>
```

```
* <tr><td>OPTIONS</td><td>{ @link #optionsForAllow }</td></tr>
```

```
* <tr><td>POST</td><td>{ @link #postForLocation }</td></tr>
```

```
* <tr><td></td><td>{ @link #postForObject }</td></tr>
```

```
* <tr><td>PUT</td><td>{ @link #put }</td></tr>
```

```
* <tr><td>any</td><td>{ @link #exchange }</td></tr>
```

```
* <tr><td></td><td>{ @link #execute }</td></tr> </table>
```

```
*
```

```
* <p>The { @code exchange } and { @code execute } methods are generalized versions of the more specific methods listed
```

```
* above them. They support additional, less frequently used combinations including support for requests using the
```

```
* HTTP PATCH method. However, note that the underlying HTTP library must also support the desired combination.
```

```
*
```

```
* <p>For each of these HTTP methods, there are three corresponding Java methods in the { @code RestTemplate }.
```

```
* Two variants take a { @code String } URI as first argument (eg. { @link #getForObject(String, Class, Object[]) },
```

```
* { @link #getForObject(String, Class, Map) }), and are capable of substituting any { @link plain UriTemplate URI templates }
```

- \* in that URL using either a `{ @code String }` variable arguments array, or a `{ @code Map<String, String> }`.
- \* The string varargs variant expands the given template variables in order, so that
- \* 

```
<pre>String result = restTemplate.getForObject("http://example.com/hotels/{hotel}/bookings/{booking}", String.class, "42", "21");
```
- \* will perform a GET on `{ @code http://example.com/hotels/42/bookings/21 }`. The map variant expands the template based
- \* on variable name, and is therefore more useful when using many variables, or when a single variable is used multiple
- \* times. For example:
- \* 

```
<pre>Map<String, String> vars = Collections.singletonMap("hotel", "42");String result = restTemplate.getForObject("http://example.com/hotels/{hotel}/rooms/{hotel}", String.class, vars);
```
- \* will perform a GET on `{ @code http://example.com/hotels/42/rooms/42 }`. Alternatively, there are `{ @link URI }` variant
- \* methods (`{ @link #getForObject(URI, Class) }`), which do not allow for URI templates, but allow you to reuse a single,
- \* expanded URI multiple times.
- \* 

Furthermore, the `{ @code String }`-argument methods assume that the URL String is unencoded. This means that
- \* 

```
<pre>restTemplate.getForObject("http://example.com/hotel list");
```
- \* will perform a GET on `{ @code http://example.com/hotel%20list }`. As a result, any URL passed that is already encoded
- \* will be encoded twice (i.e. `{ @code http://example.com/hotel%20list }` will become `{ @code http://example.com/hotel%2520list }`). If this behavior is undesirable, use the `{ @code URI }`-argument methods, which
- \* will not perform any URL encoding.
- \* 

Objects passed to and returned from these methods are converted to and from HTTP messages by `{ @link HttpMessageConverter }` instances. Converters for the main mime types are registered by default, but you can also write your own converter and register it via the `{ @link #setMessageConverters messageConverters }` bean property.
- \* 

This template uses a `{ @link org.springframework.http.client.SimpleClientHttpRequestFactory }` and a `{ @link DefaultResponseErrorHandler }` as default strategies for creating HTTP connections or handling HTTP errors, respectively. These defaults can be overridden through the `{ @link #setRequestFactory(ClientHttpRequestFactory) requestFactory }` and `{ @link #setErrorHandler(ResponseErrorHandler) errorHandler }` bean properties.
- \* 

@author Arjen Poutsma
- \* @since 3.0

```

* @see HttpMessageConverter
* @see RequestCallback
* @see ResponseExtractor
* @see ResponseErrorHandler
* @see AsyncRestTemplate
*/
public class RestTemplate extends InterceptingHttpAccessor implements RestOperations {

    private static boolean romePresent =
        ClassUtils.isPresent("com.sun.syndication.feed.WireFeed", RestTemplate.class.getClassLoader());

    private static final boolean jaxb2Present =
        ClassUtils.isPresent("javax.xml.bind.Binder", RestTemplate.class.getClassLoader());

    private static final boolean jackson2Present =
        ClassUtils.isPresent("com.fasterxml.jackson.databind.ObjectMapper", RestTemplate.class.getClassLoader()) &&
        ClassUtils.isPresent("com.fasterxml.jackson.core.JsonGenerator", RestTemplate.class.getClassLoader());

    private static final boolean jacksonPresent =
        ClassUtils.isPresent("org.codehaus.jackson.map.ObjectMapper", RestTemplate.class.getClassLoader()) &&
        ClassUtils.isPresent("org.codehaus.jackson.JsonGenerator", RestTemplate.class.getClassLoader());

    private final ResponseExtractor<HttpHeaders> headersExtractor = new HeadersExtractor();

    private List<HttpMessageConverter<?>> messageConverters = new ArrayList<HttpMessageConverter<?>>();

    private ResponseErrorHandler errorHandler = new DefaultResponseErrorHandler();

    /**
     * Create a new instance of the {@link RestTemplate} using default settings.
     */
    @SuppressWarnings("deprecation")
    public RestTemplate() {
        this.messageConverters.add(new ByteArrayHttpMessageConverter());
        this.messageConverters.add(new StringHttpMessageConverter());
        this.messageConverters.add(new ResourceHttpMessageConverter());
        this.messageConverters.add(new SourceHttpMessageConverter<Source>());
        this.messageConverters.add(new AllEncompassingFormHttpMessageConverter());
        if (romePresent) {
            this.messageConverters.add(new AtomFeedHttpMessageConverter());
            this.messageConverters.add(new RssChannelHttpMessageConverter());
        }
        if (jaxb2Present) {
            this.messageConverters.add(new Jaxb2RootElementHttpMessageConverter());
        }
        if (jackson2Present) {

```

```

    this.messageConverters.add(new MappingJackson2HttpMessageConverter());
}
else if (jacksonPresent) {
    this.messageConverters.add(new
org.springframework.http.converter.json.MappingJacksonHttpMessageConverter());
}
}

/**
 * Create a new instance of the {@link RestTemplate} based on the given {@link ClientHttpRequestFactory}.
 * @param requestFactory HTTP request factory to use
 * @see org.springframework.http.client.SimpleClientHttpRequestFactory
 * @see org.springframework.http.client.HttpComponentsClientHttpRequestFactory
 */
public RestTemplate(ClientHttpRequestFactory requestFactory) {
    this();
    setRequestFactory(requestFactory);
}

/**
 * Set the message body converters to use.
 * <p>These converters are used to convert from and to HTTP requests and responses.
 */
public void setMessageConverters(List<HttpMessageConverter<?>> messageConverters) {
    Assert.notEmpty(messageConverters, "'messageConverters' must not be empty");
    this.messageConverters = messageConverters;
}

/**
 * Return the message body converters.
 */
public List<HttpMessageConverter<?>> getMessageConverters() {
    return this.messageConverters;
}

/**
 * Set the error handler.
 * <p>By default, RestTemplate uses a {@link DefaultResponseErrorHandler}.
 */
public void setErrorHandler(ResponseErrorHandler errorHandler) {
    Assert.notNull(errorHandler, "'errorHandler' must not be null");
    this.errorHandler = errorHandler;
}

/**
 * Return the error handler.
 */

```

```

public ResponseErrorHandler getErrorHandler() {
    return this.errorHandler;
}

// GET

@Override
public <T> T getForObject(String url, Class<T> responseType, Object... urlVariables) throws RestClientException
{
    RequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    HttpResponseMessageConverterExtractor<T> responseExtractor =
        new HttpResponseMessageConverterExtractor<T>(responseType, getMessageConverters(), logger);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor, urlVariables);
}

@Override
public <T> T getForObject(String url, Class<T> responseType, Map<String, ?> urlVariables) throws
RestClientException {
    RequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    HttpResponseMessageConverterExtractor<T> responseExtractor =
        new HttpResponseMessageConverterExtractor<T>(responseType, getMessageConverters(), logger);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor, urlVariables);
}

@Override
public <T> T getForObject(Uri url, Class<T> responseType) throws RestClientException {
    RequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    HttpResponseMessageConverterExtractor<T> responseExtractor =
        new HttpResponseMessageConverterExtractor<T>(responseType, getMessageConverters(), logger);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor);
}

@Override
public <T> ResponseEntity<T> getForEntity(String url, Class<T> responseType, Object... urlVariables)
throws RestClientException {
    RequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor, urlVariables);
}

@Override
public <T> ResponseEntity<T> getForEntity(String url, Class<T> responseType, Map<String, ?> urlVariables)
throws RestClientException {
    RequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor, urlVariables);
}

```

```

@Override
public <T> ResponseEntity<T> getForEntity(Uri url, Class<T> responseType) throws RestClientException {
    RequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor);
}

```

```
// HEAD
```

```

@Override
public HttpHeaders headForHeaders(String url, Object... urlVariables) throws RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    return execute(url, HttpMethod.HEAD, null, headersExtractor, urlVariables);
}

```

```

@Override
public HttpHeaders headForHeaders(String url, Map<String, ?> urlVariables) throws RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    return execute(url, HttpMethod.HEAD, null, headersExtractor, urlVariables);
}

```

```

@Override
public HttpHeaders headForHeaders(Uri url) throws RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    return execute(url, HttpMethod.HEAD, null, headersExtractor);
}

```

```
// POST
```

```

@Override
public Uri postForLocation(String url, Object request, Object... urlVariables) throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request);
    HttpHeaders headers = execute(url, HttpMethod.POST, requestCallback, this.headersExtractor, urlVariables);
    return headers.getLocation();
}

```

```

@Override
public Uri postForLocation(String url, Object request, Map<String, ?> urlVariables)
    throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request);
    HttpHeaders headers = execute(url, HttpMethod.POST, requestCallback, this.headersExtractor, urlVariables);
    return headers.getLocation();
}

```

```

@Override
public Uri postForLocation(Uri url, Object request) throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request);
}

```



```

HttpHeaders headers = execute(url, HttpMethod.POST, requestCallback, this.headersExtractor);
return headers.getLocation();
}

```

```

@Override
public <T> T postForObject(String url, Object request, Class<T> responseType, Object... uriVariables)
    throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request, responseType);
    HttpMessageConverterExtractor<T> responseExtractor =
        new HttpMessageConverterExtractor<T>(responseType, getMessageConverters(), logger);
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor, uriVariables);
}

```

```

@Override
public <T> T postForObject(String url, Object request, Class<T> responseType, Map<String, ?> uriVariables)
    throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request, responseType);
    HttpMessageConverterExtractor<T> responseExtractor =
        new HttpMessageConverterExtractor<T>(responseType, getMessageConverters(), logger);
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor, uriVariables);
}

```

```

@Override
public <T> T postForObject(URI url, Object request, Class<T> responseType) throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request, responseType);
    HttpMessageConverterExtractor<T> responseExtractor =
        new HttpMessageConverterExtractor<T>(responseType, getMessageConverters());
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor);
}

```

```

@Override
public <T> ResponseEntity<T> postForEntity(String url, Object request, Class<T> responseType, Object...
uriVariables)
    throws RestClientException {

    RequestCallback requestCallback = httpEntityCallback(request, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor, uriVariables);
}

```

```

@Override
public <T> ResponseEntity<T> postForEntity(String url, Object request, Class<T> responseType, Map<String, ?>
uriVariables)
    throws RestClientException {

    RequestCallback requestCallback = httpEntityCallback(request, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor, uriVariables);
}

```

```

}

@Override
public <T> ResponseEntity<T> postForEntity(URI url, Object request, Class<T> responseType) throws
RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor);
}

// PUT

@Override
public void put(String url, Object request, Object... urlVariables) throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request);
    execute(url, HttpMethod.PUT, requestCallback, null, urlVariables);
}

@Override
public void put(String url, Object request, Map<String, ?> urlVariables) throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request);
    execute(url, HttpMethod.PUT, requestCallback, null, urlVariables);
}

@Override
public void put(URI url, Object request) throws RestClientException {
    RequestCallback requestCallback = httpEntityCallback(request);
    execute(url, HttpMethod.PUT, requestCallback, null);
}

// DELETE

@Override
public void delete(String url, Object... urlVariables) throws RestClientException {
    execute(url, HttpMethod.DELETE, null, null, urlVariables);
}

@Override
public void delete(String url, Map<String, ?> urlVariables) throws RestClientException {
    execute(url, HttpMethod.DELETE, null, null, urlVariables);
}

@Override
public void delete(URI url) throws RestClientException {
    execute(url, HttpMethod.DELETE, null, null);
}

// OPTIONS

```

```

@Override
public Set<HttpMethod> optionsForAllow(String url, Object... uriVariables) throws RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    HttpHeaders headers = execute(url, HttpMethod.OPTIONS, null, headersExtractor, uriVariables);
    return headers.getAllow();
}

```

```

@Override
public Set<HttpMethod> optionsForAllow(String url, Map<String, ?> uriVariables) throws RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    HttpHeaders headers = execute(url, HttpMethod.OPTIONS, null, headersExtractor, uriVariables);
    return headers.getAllow();
}

```

```

@Override
public Set<HttpMethod> optionsForAllow(URI url) throws RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    HttpHeaders headers = execute(url, HttpMethod.OPTIONS, null, headersExtractor);
    return headers.getAllow();
}

```

```
// exchange
```

```

@Override
public <T> ResponseEntity<T> exchange(String url, HttpMethod method,
    HttpEntity<?> requestEntity, Class<T> responseType, Object... uriVariables) throws RestClientException {

    RequestCallback requestCallback = httpEntityCallback(requestEntity, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, method, requestCallback, responseExtractor, uriVariables);
}

```

```

@Override
public <T> ResponseEntity<T> exchange(String url, HttpMethod method,
    HttpEntity<?> requestEntity, Class<T> responseType, Map<String, ?> uriVariables) throws RestClientException {

    RequestCallback requestCallback = httpEntityCallback(requestEntity, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, method, requestCallback, responseExtractor, uriVariables);
}

```

```

@Override
public <T> ResponseEntity<T> exchange(URI url, HttpMethod method, HttpEntity<?> requestEntity,
    Class<T> responseType) throws RestClientException {

    RequestCallback requestCallback = httpEntityCallback(requestEntity, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
}

```

```

return execute(url, method, requestCallback, responseExtractor);
}

@Override
public <T> ResponseEntity<T> exchange(String url, HttpMethod method, HttpEntity<?> requestEntity,
    ParameterizedTypeReference<T> responseType, Object... uriVariables) throws RestClientException {

    Type type = responseType.getType();
    RequestCallback requestCallback = httpEntityCallback(requestEntity, type);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(type);
    return execute(url, method, requestCallback, responseExtractor, uriVariables);
}

@Override
public <T> ResponseEntity<T> exchange(String url, HttpMethod method, HttpEntity<?> requestEntity,
    ParameterizedTypeReference<T> responseType, Map<String, ?> uriVariables) throws RestClientException {

    Type type = responseType.getType();
    RequestCallback requestCallback = httpEntityCallback(requestEntity, type);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(type);
    return execute(url, method, requestCallback, responseExtractor, uriVariables);
}

@Override
public <T> ResponseEntity<T> exchange(URI url, HttpMethod method, HttpEntity<?> requestEntity,
    ParameterizedTypeReference<T> responseType) throws RestClientException {

    Type type = responseType.getType();
    RequestCallback requestCallback = httpEntityCallback(requestEntity, type);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(type);
    return execute(url, method, requestCallback, responseExtractor);
}

// general execution

@Override
public <T> T execute(String url, HttpMethod method, RequestCallback requestCallback,
    ResponseExtractor<T> responseExtractor, Object... uriVariables) throws RestClientException {

    URI expanded = new UriTemplate(url).expand(uriVariables);
    return doExecute(expanded, method, requestCallback, responseExtractor);
}

@Override
public <T> T execute(String url, HttpMethod method, RequestCallback requestCallback,
    ResponseExtractor<T> responseExtractor, Map<String, ?> uriVariables) throws RestClientException {

    URI expanded = new UriTemplate(url).expand(uriVariables);

```

```

return doExecute(expanded, method, requestCallback, responseExtractor);
}

@Override
public <T> T execute(URI url, HttpMethod method, RequestCallback requestCallback,
    ResponseExtractor<T> responseExtractor) throws RestClientException {

    return doExecute(url, method, requestCallback, responseExtractor);
}

/**
 * Execute the given method on the provided URI. The {@link ClientHttpRequest} is processed using the {@link
 * RequestCallback}; the response with the {@link ResponseExtractor}.
 * @param url the fully-expanded URL to connect to
 * @param method the HTTP method to execute (GET, POST, etc.)
 * @param requestCallback object that prepares the request (can be {@code null})
 * @param responseExtractor object that extracts the return value from the response (can be {@code null})
 * @return an arbitrary object, as returned by the {@link ResponseExtractor}
 */
protected <T> T doExecute(URI url, HttpMethod method, RequestCallback requestCallback,
    ResponseExtractor<T> responseExtractor) throws RestClientException {

    Assert.notNull(url, "'url' must not be null");
    Assert.notNull(method, "'method' must not be null");
    ClientHttpResponse response = null;
    try {
        ClientHttpRequest request = createRequest(url, method);
        if (requestCallback != null) {
            requestCallback.doWithRequest(request);
        }
        response = request.execute();
        if (!getErrorHandler().hasError(response)) {
            logResponseStatus(method, url, response);
        }
        else {
            handleResponseError(method, url, response);
        }
        if (responseExtractor != null) {
            return responseExtractor.extractData(response);
        }
        else {
            return null;
        }
    }
    catch (IOException ex) {
        throw new ResourceAccessException("I/O error on " + method.name() +
            " request for \"" + url + "\": " + ex.getMessage(), ex);
    }
}

```

```

finally {
    if (response != null) {
        response.close();
    }
}

private void logResponseStatus(HttpMethod method, URI url, ClientHttpResponse response) {
    if (logger.isDebugEnabled()) {
        try {
            logger.debug(method.name() + " request for \"" + url + "\" resulted in " +
                response.getStatusCode() + " (" +
                response.getStatusText() + ")");
        }
        catch (IOException e) {
            // ignore
        }
    }
}

private void handleResponseError(HttpMethod method, URI url, ClientHttpResponse response) throws IOException
{
    if (logger.isWarnEnabled()) {
        try {
            logger.warn(
                method.name() + " request for \"" + url + "\" resulted in " + response.getStatusCode() + " (" +
                response.getStatusText() + "); invoking error handler");
        }
        catch (IOException e) {
            // ignore
        }
    }
    getErrorHandler().handleError(response);
}

/**
 * Returns a request callback implementation that prepares the request {@code Accept}
 * headers based on the given response type and configured
 * {@code @linkplain #getMessageConverters()} message converters}.
 */
protected <T> RequestCallback acceptHeaderRequestCallback(Class<T> responseType) {
    return new AcceptHeaderRequestCallback(responseType);
}

/**
 * Returns a request callback implementation that writes the given object to the
 * request stream.
 */

```

```

protected <T> RequestCallback httpEntityCallback(Object requestBody) {
    return new HttpEntityRequestCallback(requestBody);
}

/**
 * Returns a request callback implementation that writes the given object to the
 * request stream.
 */
protected <T> RequestCallback httpEntityCallback(Object requestBody,
    Type responseType) {
    return new HttpEntityRequestCallback(requestBody, responseType);
}

/**
 * Returns a response extractor for { @link ResponseEntity }.
 */
protected <T> ResponseExtractor<ResponseEntity<T>> responseEntityExtractor(Type responseType) {
    return new ResponseEntityResponseExtractor<T>(responseType);
}

/**
 * Returns a response extractor for { @link HttpHeaders }.
 */
protected ResponseExtractor<HttpHeaders> headersExtractor() {
    return this.headersExtractor;
}

/**
 * Request callback implementation that prepares the request's accept headers.
 */
private class AcceptHeaderRequestCallback implements RequestCallback {

    private final Type responseType;

    private AcceptHeaderRequestCallback(Type responseType) {
        this.responseType = responseType;
    }

    @Override
    public void doWithRequest(ClientHttpRequest request) throws IOException {
        if (responseType != null) {
            Class<?> responseClass = null;
            if (responseType instanceof Class) {
                responseClass = (Class<?>) responseType;
            }

            List<MediaType> allSupportedMediaTypes = new ArrayList<MediaType>();

```

```

for (HttpMessageConverter<?> converter : getMessageConverters()) {
    if (responseClass != null) {
        if (converter.canRead(responseClass, null)) {
            allSupportedMediaTypes.addAll(getSupportedMediaTypes(converter));
        }
    }
    else if (converter instanceof GenericHttpMessageConverter) {

        GenericHttpMessageConverter<?> genericConverter = (GenericHttpMessageConverter<?>) converter;
        if (genericConverter.canRead(responseType, null, null)) {
            allSupportedMediaTypes.addAll(getSupportedMediaTypes(converter));
        }
    }
}

if (!allSupportedMediaTypes.isEmpty()) {
    MediaType.sortBySpecificity(allSupportedMediaTypes);
    if (logger.isDebugEnabled()) {
        logger.debug("Setting request Accept header to " +
            allSupportedMediaTypes);
    }
    request.getHeaders().setAccept(allSupportedMediaTypes);
}
}
}

private List<MediaType> getSupportedMediaTypes(HttpMessageConverter<?> messageConverter) {
    List<MediaType> supportedMediaTypes = messageConverter.getSupportedMediaTypes();
    List<MediaType> result = new ArrayList<MediaType>(supportedMediaTypes.size());
    for (MediaType supportedMediaType : supportedMediaTypes) {
        if (supportedMediaType.getCharSet() != null) {
            supportedMediaType =
                new MediaType(supportedMediaType.getType(), supportedMediaType.getSubtype());
        }
        result.add(supportedMediaType);
    }
    return result;
}

/**
 * Request callback implementation that writes the given object to the request stream.
 */
private class HttpEntityRequestCallback extends AcceptHeaderRequestCallback {

    private final HttpEntity<?> requestEntity;

```



```

private HttpEntityRequestCallback(Object requestBody) {
    this(requestBody, null);
}

private HttpEntityRequestCallback(Object requestBody, Type responseType) {
    super(responseType);
    if (requestBody instanceof HttpEntity) {
        this.requestEntity = (HttpEntity<?>) requestBody;
    }
    else if (requestBody != null) {
        this.requestEntity = new HttpEntity<Object>(requestBody);
    }
    else {
        this.requestEntity = HttpEntity.EMPTY;
    }
}

@Override
@SuppressWarnings("unchecked")
public void doWithRequest(ClientHttpRequest httpRequest) throws IOException {
    super.doWithRequest(httpRequest);
    if (!requestEntity.hasBody()) {
        HttpHeaders httpHeaders = httpRequest.getHeaders();
        HttpHeaders requestHeaders = requestEntity.getHeaders();
        if (!requestHeaders.isEmpty()) {
            httpHeaders.putAll(requestHeaders);
        }
        if (httpHeaders.getContentLength() == -1) {
            httpHeaders.setContentLength(0L);
        }
    }
    else {
        Object requestBody = requestEntity.getBody();
        Class<?> requestType = requestBody.getClass();
        HttpHeaders requestHeaders = requestEntity.getHeaders();
        MediaType requestContentType = requestHeaders.getContentType();
        for (HttpMessageConverter<?> messageConverter : getMessageConverters()) {
            if (messageConverter.canWrite(requestType, requestContentType)) {
                if (!requestHeaders.isEmpty()) {
                    httpRequest.getHeaders().putAll(requestHeaders);
                }
                if (logger.isDebugEnabled()) {
                    if (requestContentType != null) {
                        logger.debug("Writing [" + requestBody + "] as \"" + requestContentType +
                            "\" using [" + messageConverter + "]");
                    }
                }
                else {
                    logger.debug("Writing [" + requestBody + "] using [" + messageConverter + "]");
                }
            }
        }
    }
}

```

```

    }

    }
    ((HttpMessageConverter<Object>) messageConverter).write(
        requestBody, requestContentType, httpRequest);
    return;
    }
    }
    String message = "Could not write request: no suitable HttpMessageConverter found for request type [" +
        requestType.getName() + "]";
    if (requestContentType != null) {
        message += " and content type [" + requestContentType + "]";
    }
    throw new RestClientException(message);
    }
    }
    }

/**
 * Response extractor for {@link HttpEntity}.
 */
private class ResponseEntityResponseExtractor<T> implements ResponseExtractor<ResponseEntity<T>> {

    private final HttpMessageConverterExtractor<T> delegate;

    public ResponseEntityResponseExtractor(Type responseType) {
        if (responseType != null && !Void.class.equals(responseType)) {
            this.delegate = new HttpMessageConverterExtractor<T>(responseType, getMessageConverters(), logger);
        } else {
            this.delegate = null;
        }
    }

    @Override
    public ResponseEntity<T> extractData(ClientHttpResponse response) throws IOException {
        if (this.delegate != null) {
            T body = this.delegate.extractData(response);
            return new ResponseEntity<T>(body, response.getHeaders(), response.getStatusCode());
        }
        else {
            return new ResponseEntity<T>(response.getHeaders(), response.getStatusCode());
        }
    }
}

/**

```

```

* Response extractor that extracts the response { @link HttpHeaders}.
*/
private static class HeadersExtractor implements ResponseExtractor<HttpHeaders> {

    @Override
    public HttpHeaders extractData(ClientHttpResponse response) throws IOException {
        return response.getHeaders();
    }
}

}
/*
* Copyright 2002-2013 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package org.springframework.web.client;

import java.io.IOException;
import java.io.OutputStream;
import java.lang.reflect.Type;
import java.net.URI;
import java.util.List;
import java.util.Map;
import java.util.Set;
import java.util.concurrent.ExecutionException;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.TimeoutException;

import org.springframework.core.ParameterizedTypeReference;
import org.springframework.core.task.AsyncListenableTaskExecutor;
import org.springframework.core.task.AsyncTaskExecutor;
import org.springframework.core.task.SimpleAsyncTaskExecutor;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.http.ResponseEntity;

```

```

import org.springframework.http.client.AsyncClientHttpRequest;
import org.springframework.http.client.AsyncClientHttpRequestFactory;
import org.springframework.http.client.ClientHttpRequest;
import org.springframework.http.client.ClientHttpRequestFactory;
import org.springframework.http.client.ClientHttpResponse;
import org.springframework.http.client.SimpleClientHttpRequestFactory;
import org.springframework.http.client.support.AsyncHttpAccessor;
import org.springframework.http.converter.HttpMessageConverter;
import org.springframework.util.Assert;
import org.springframework.util.concurrent.ListenableFuture;
import org.springframework.util.concurrent.ListenableFutureAdapter;
import org.springframework.util.concurrent.ListenableFutureCallback;
import org.springframework.web.util.UriTemplate;

/**
 * <strong>Spring's central class for asynchronous client-side HTTP access.</strong>
 * Exposes similar methods as { @link RestTemplate }, but returns { @link ListenableFuture }
 * wrappers as opposed to concrete results.
 *
 * <p>The { @code AsyncRestTemplate } exposes a synchronous { @link RestTemplate } via the
 * { @link #getRestOperations() } method, and it shares its
 * { @linkplain #setErrorHandler(ResponseErrorHandler) } error handler } and
 * { @linkplain #setMessageConverters(List) } message converters } with this
 * { @code RestTemplate }.
 *
 * <p>For more information, please refer to the { @link RestTemplate } API documentation.
 *
 * @author Arjen Poutsma
 * @since 4.0
 * @see RestTemplate
 */
public class AsyncRestTemplate extends AsyncHttpAccessor implements AsyncRestOperations {

    private final RestTemplate syncTemplate;

    /**
     * Create a new instance of the { @code AsyncRestTemplate } using default settings.
     * <p>This constructor uses a { @link SimpleClientHttpRequestFactory } in combination
     * with a { @link SimpleAsyncTaskExecutor } for asynchronous execution.
     */
    public AsyncRestTemplate() {
        this(new SimpleAsyncTaskExecutor());
    }

    /**
     * Create a new instance of the { @code AsyncRestTemplate } using the given
     * { @link AsyncTaskExecutor }.

```

```

* <p>This constructor uses a { @link SimpleClientHttpRequestFactory } in combination
* with the given { @code AsyncTaskExecutor } for asynchronous execution.
*/
public AsyncRestTemplate(AsyncListenableTaskExecutor taskExecutor) {
    Assert.notNull(taskExecutor, "AsyncTaskExecutor must not be null");
    SimpleClientHttpRequestFactory requestFactory =
        new SimpleClientHttpRequestFactory();
    requestFactory.setTaskExecutor(taskExecutor);
    this.syncTemplate = new RestTemplate(requestFactory);
    setAsyncRequestFactory(requestFactory);
}

/**
* Create a new instance of the { @code AsyncRestTemplate } using the given
* { @link AsyncClientHttpRequestFactory }.
* <p>This constructor will cast the given asynchronous
* { @code AsyncClientHttpRequestFactory } to a { @link ClientHttpRequestFactory }. Since
* all implementations of { @code ClientHttpRequestFactory } provided in Spring also
* implement { @code AsyncClientHttpRequestFactory }, this should not result in a
* { @code ClassCastException }.
*/
public AsyncRestTemplate(AsyncClientHttpRequestFactory asyncRequestFactory) {
    this(asyncRequestFactory, (ClientHttpRequestFactory) asyncRequestFactory);
}

/**
* Creates a new instance of the { @code AsyncRestTemplate } using the given
* asynchronous and synchronous request factories.
* @param asyncRequestFactory the asynchronous request factory
* @param syncRequestFactory the synchronous request factory
*/
public AsyncRestTemplate(AsyncClientHttpRequestFactory asyncRequestFactory,
    ClientHttpRequestFactory syncRequestFactory) {
    this(asyncRequestFactory, new RestTemplate(syncRequestFactory));
}

/**
* Create a new instance of the { @code AsyncRestTemplate } using the given
* { @link AsyncClientHttpRequestFactory } and synchronous { @link RestTemplate }.
* @param requestFactory the asynchronous request factory to use
* @param restTemplate the synchronous template to use
*/
public AsyncRestTemplate(AsyncClientHttpRequestFactory requestFactory,
    RestTemplate restTemplate) {
    Assert.notNull(restTemplate, "'restTemplate' must not be null");
    this.syncTemplate = restTemplate;
    setAsyncRequestFactory(requestFactory);
}

```

```

/**
 * Set the error handler.
 * <p>By default, AsyncRestTemplate uses a
 * { @link org.springframework.web.client.DefaultResponseErrorHandler}.
 */
public void setErrorHandler(ResponseErrorHandler errorHandler) {
    this.syncTemplate.setErrorHandler(errorHandler);
}

/** Return the error handler. */
public ResponseErrorHandler getErrorHandler() {
    return this.syncTemplate.getErrorHandler();
}

@Override
public RestOperations getRestOperations() {
    return this.syncTemplate;
}

/**
 * Set the message body converters to use.
 * <p>These converters are used to convert from and to HTTP requests and responses.
 */
public void setMessageConverters(List<HttpMessageConverter<?>> messageConverters) {
    this.syncTemplate.setMessageConverters(messageConverters);
}

/**
 * Return the message body converters.
 */
public List<HttpMessageConverter<?>> getMessageConverters() {
    return syncTemplate.getMessageConverters();
}

// GET

@Override
public <T> ListenableFuture<ResponseEntity<T>> getForEntity(String url, Class<T> responseType, Object...
uriVariables)
    throws RestClientException {
    AsyncRequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor, uriVariables);
}

```

```

@Override
public <T> ListenableFuture<ResponseEntity<T>> getForEntity(String url, Class<T> responseType,
    Map<String, ?> urlVariables) throws RestClientException {

    AsyncRequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor, urlVariables);
}

@Override
public <T> ListenableFuture<ResponseEntity<T>> getForEntity(URI url, Class<T> responseType) throws
RestClientException {
    AsyncRequestCallback requestCallback = acceptHeaderRequestCallback(responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor = responseEntityExtractor(responseType);
    return execute(url, HttpMethod.GET, requestCallback, responseExtractor);
}

// HEAD

@Override
public ListenableFuture<HttpHeaders> headForHeaders(String url, Object... uriVariables) throws
RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    return execute(url, HttpMethod.HEAD, null, headersExtractor, uriVariables);
}

@Override
public ListenableFuture<HttpHeaders> headForHeaders(String url, Map<String, ?> uriVariables) throws
RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    return execute(url, HttpMethod.HEAD, null, headersExtractor, uriVariables);
}

@Override
public ListenableFuture<HttpHeaders> headForHeaders(URI url) throws RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    return execute(url, HttpMethod.HEAD, null, headersExtractor);
}

// POST

@Override
public ListenableFuture<URI> postForLocation(String url, HttpEntity<?> request,
    Object... uriVariables) throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request);
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    ListenableFuture<HttpHeaders> headersFuture =

```

```

        execute(url, HttpMethod.POST, requestCallback, headersExtractor,
            uriVariables);
    return extractLocationHeader(headersFuture);
}

@Override
public ListenableFuture<URI> postForLocation(String url, HttpEntity<?> request,
    Map<String, ?> uriVariables) throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request);
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    ListenableFuture<HttpHeaders> headersFuture =
        execute(url, HttpMethod.POST, requestCallback, headersExtractor,
            uriVariables);
    return extractLocationHeader(headersFuture);
}

@Override
public ListenableFuture<URI> postForLocation(URI url, HttpEntity<?> request)
    throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request);
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    ListenableFuture<HttpHeaders> headersFuture =
        execute(url, HttpMethod.POST, requestCallback, headersExtractor);
    return extractLocationHeader(headersFuture);
}

private static ListenableFuture<URI> extractLocationHeader(final ListenableFuture<HttpHeaders> headersFuture) {
    return new ListenableFuture<URI>() {

        @Override
        public void addCallback(final ListenableFutureCallback<? super URI> callback) {
            headersFuture.addCallback(new ListenableFutureCallback<HttpHeaders>() {
                @Override
                public void onSuccess(HttpHeaders result) {
                    callback.onSuccess(result.getLocation());
                }
            });
        }

        @Override
        public void onFailure(Throwable t) {
            callback.onFailure(t);
        }
    });
}

@Override
public boolean cancel(boolean mayInterruptIfRunning) {
    return headersFuture.cancel(mayInterruptIfRunning);
}

```



```

@Override
public boolean isCancelled() {
    return headersFuture.isCancelled();
}
@Override
public boolean isDone() {
    return headersFuture.isDone();
}
@Override
public URI get() throws InterruptedException, ExecutionException {
    HttpHeaders headers = headersFuture.get();
    return headers.getLocation();
}
@Override
public URI get(long timeout, TimeUnit unit)
    throws InterruptedException, ExecutionException, TimeoutException {
    HttpHeaders headers = headersFuture.get(timeout, unit);
    return headers.getLocation();
}
};
}

@Override
public <T> ListenableFuture<ResponseEntity<T>> postForEntity(String url, HttpEntity<?> request,
    Class<T> responseType, Object... uriVariables) throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =
        responseEntityExtractor(responseType);
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor,
        uriVariables);
}

@Override
public <T> ListenableFuture<ResponseEntity<T>> postForEntity(String url, HttpEntity<?> request,
    Class<T> responseType, Map<String, ?> uriVariables)
    throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =
        responseEntityExtractor(responseType);
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor,
        uriVariables);
}

@Override
public <T> ListenableFuture<ResponseEntity<T>> postForEntity(URI url, HttpEntity<?> request,
    Class<T> responseType) throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =

```

```

    responseEntityExtractor(responseType);
    return execute(url, HttpMethod.POST, requestCallback, responseExtractor);
}

// PUT

@Override
public ListenableFuture<?> put(String url, HttpEntity<?> request, Object... uriVariables)
    throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request);
    return execute(url, HttpMethod.PUT, requestCallback, null, uriVariables);
}

@Override
public ListenableFuture<?> put(String url, HttpEntity<?> request,
    Map<String, ?> uriVariables) throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request);
    return execute(url, HttpMethod.PUT, requestCallback, null, uriVariables);
}

@Override
public ListenableFuture<?> put(URI url, HttpEntity<?> request) throws RestClientException {
    AsyncRequestCallback requestCallback = httpEntityCallback(request);
    return execute(url, HttpMethod.PUT, requestCallback, null);
}

// DELETE

@Override
public ListenableFuture<?> delete(String url, Object... urlVariables)
    throws RestClientException {
    return execute(url, HttpMethod.DELETE, null, null, urlVariables);
}

@Override
public ListenableFuture<?> delete(String url, Map<String, ?> urlVariables)
    throws RestClientException {
    return execute(url, HttpMethod.DELETE, null, null, urlVariables);
}

@Override
public ListenableFuture<?> delete(URI url) throws RestClientException {
    return execute(url, HttpMethod.DELETE, null, null);
}

// OPTIONS

@Override

```

```

public ListenableFuture<Set<HttpMethod>> optionsForAllow(String url, Object... uriVariables) throws
RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    ListenableFuture<HttpHeaders> headersFuture = execute(url, HttpMethod.OPTIONS, null, headersExtractor,
uriVariables);
    return extractAllowHeader(headersFuture);
}

```

@Override

```

public ListenableFuture<Set<HttpMethod>> optionsForAllow(String url, Map<String, ?> uriVariables) throws
RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    ListenableFuture<HttpHeaders> headersFuture = execute(url, HttpMethod.OPTIONS, null, headersExtractor,
uriVariables);
    return extractAllowHeader(headersFuture);
}

```

@Override

```

public ListenableFuture<Set<HttpMethod>> optionsForAllow(URI url) throws RestClientException {
    ResponseExtractor<HttpHeaders> headersExtractor = headersExtractor();
    ListenableFuture<HttpHeaders> headersFuture = execute(url, HttpMethod.OPTIONS, null, headersExtractor);
    return extractAllowHeader(headersFuture);
}

```

```

private static ListenableFuture<Set<HttpMethod>> extractAllowHeader(final ListenableFuture<HttpHeaders>
headersFuture) {
    return new ListenableFuture<Set<HttpMethod>>() {

```

@Override

```

public void addCallback(
    final ListenableFutureCallback<? super Set<HttpMethod>> callback) {
    headersFuture.addCallback(new ListenableFutureCallback<HttpHeaders>() {
        @Override
        public void onSuccess(HttpHeaders result) {
            callback.onSuccess(result.getAllow());
        }
    });
}

```

@Override

```

public void onFailure(Throwable t) {
    callback.onFailure(t);
}
});
}

```

@Override

```

public boolean cancel(boolean mayInterruptIfRunning) {
    return headersFuture.cancel(mayInterruptIfRunning);
}

```

```

@Override
public boolean isCancelled() {
    return headersFuture.isCancelled();
}
@Override
public boolean isDone() {
    return headersFuture.isDone();
}
@Override
public Set<HttpMethod> get() throws InterruptedException, ExecutionException {
    HttpHeaders headers = headersFuture.get();
    return headers.getAllow();
}
@Override
public Set<HttpMethod> get(long timeout, TimeUnit unit)
    throws InterruptedException, ExecutionException, TimeoutException {
    HttpHeaders headers = headersFuture.get(timeout, unit);
    return headers.getAllow();
}
};
}

// exchange

@Override
public <T> ListenableFuture<ResponseEntity<T>> exchange(String url, HttpMethod method,
    HttpEntity<?> requestEntity, Class<T> responseType, Object... uriVariables)
    throws RestClientException {
    AsyncRequestCallback requestCallback =
        httpEntityCallback(requestEntity, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =
        responseEntityExtractor(responseType);
    return execute(url, method, requestCallback, responseExtractor, uriVariables);
}

@Override
public <T> ListenableFuture<ResponseEntity<T>> exchange(String url, HttpMethod method,
    HttpEntity<?> requestEntity, Class<T> responseType,
    Map<String, ?> uriVariables) throws RestClientException {
    AsyncRequestCallback requestCallback =
        httpEntityCallback(requestEntity, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =
        responseEntityExtractor(responseType);
    return execute(url, method, requestCallback, responseExtractor, uriVariables);
}

@Override

```

```

public <T> ListenableFuture<ResponseEntity<T>> exchange(URI url, HttpMethod method,
    HttpEntity<?> requestEntity, Class<T> responseType)
    throws RestClientException {
    AsyncRequestCallback requestCallback =
        httpEntityCallback(requestEntity, responseType);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =
        responseEntityExtractor(responseType);
    return execute(url, method, requestCallback, responseExtractor);
}

```

@Override

```

public <T> ListenableFuture<ResponseEntity<T>> exchange(String url, HttpMethod method,
    HttpEntity<?> requestEntity, ParameterizedTypeReference<T> responseType,
    Object... uriVariables) throws RestClientException {
    Type type = responseType.getType();
    AsyncRequestCallback requestCallback = httpEntityCallback(requestEntity, type);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =
        responseEntityExtractor(type);
    return execute(url, method, requestCallback, responseExtractor, uriVariables);
}

```

@Override

```

public <T> ListenableFuture<ResponseEntity<T>> exchange(String url, HttpMethod method,
    HttpEntity<?> requestEntity, ParameterizedTypeReference<T> responseType,
    Map<String, ?> uriVariables) throws RestClientException {
    Type type = responseType.getType();
    AsyncRequestCallback requestCallback = httpEntityCallback(requestEntity, type);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =
        responseEntityExtractor(type);
    return execute(url, method, requestCallback, responseExtractor, uriVariables);
}

```

@Override

```

public <T> ListenableFuture<ResponseEntity<T>> exchange(URI url, HttpMethod method,
    HttpEntity<?> requestEntity, ParameterizedTypeReference<T> responseType)
    throws RestClientException {
    Type type = responseType.getType();
    AsyncRequestCallback requestCallback = httpEntityCallback(requestEntity, type);
    ResponseExtractor<ResponseEntity<T>> responseExtractor =
        responseEntityExtractor(type);
    return execute(url, method, requestCallback, responseExtractor);
}

```

// general execution

@Override

```

public <T> ListenableFuture<T> execute(String url, HttpMethod method,

```

```

    AsyncRequestCallback requestCallback, ResponseExtractor<T> responseExtractor,
    Object... urlVariables) throws RestClientException {

    URI expanded = new UriTemplate(url).expand(urlVariables);
    return doExecute(expanded, method, requestCallback, responseExtractor);
}

@Override
public <T> ListenableFuture<T> execute(String url, HttpMethod method,
    AsyncRequestCallback requestCallback, ResponseExtractor<T> responseExtractor,
    Map<String, ?> urlVariables) throws RestClientException {

    URI expanded = new UriTemplate(url).expand(urlVariables);
    return doExecute(expanded, method, requestCallback, responseExtractor);
}

@Override
public <T> ListenableFuture<T> execute(URI url, HttpMethod method,
    AsyncRequestCallback requestCallback, ResponseExtractor<T> responseExtractor)
    throws RestClientException {

    return doExecute(url, method, requestCallback, responseExtractor);
}

/**
 * Execute the given method on the provided URI. The
 * { @link org.springframework.http.client.ClientHttpRequest }
 * is processed using the { @link RequestCallback }; the response with
 * the { @link ResponseExtractor }.
 * @param url the fully-expanded URL to connect to
 * @param method the HTTP method to execute (GET, POST, etc.)
 * @param requestCallback object that prepares the request (can be { @code null })
 * @param responseExtractor object that extracts the return value from the response (can
 * be { @code null })
 * @return an arbitrary object, as returned by the { @link ResponseExtractor }
 */
protected <T> ListenableFuture<T> doExecute(URI url, HttpMethod method, AsyncRequestCallback
requestCallback,
    ResponseExtractor<T> responseExtractor) throws RestClientException {

    Assert.notNull(url, "'url' must not be null");
    Assert.notNull(method, "'method' must not be null");
    try {
        AsyncClientHttpRequest request = createAsyncRequest(url, method);
        if (requestCallback != null) {
            requestCallback.doWithRequest(request);
        }
        ListenableFuture<ClientHttpResponse> responseFuture = request.executeAsync();

```

```

return new ResponseExtractorFuture<T>(method, url, responseFuture,
    responseExtractor);
}
catch (IOException ex) {
    throw new ResourceAccessException("I/O error on " + method.name() +
        " request for \"" + url + "\":" + ex.getMessage(), ex);
}
}

private void logResponseStatus(HttpMethod method, URI url, ClientHttpResponse response) {
    if (logger.isDebugEnabled()) {
        try {
            logger.debug("Async " + method.name() + " request for \"" + url +
                "\" resulted in " + response.getStatusCode() + " (" +
                    response.getStatusText() + ")");
        }
        catch (IOException ex) {
            // ignore
        }
    }
}

private void handleResponseError(HttpMethod method, URI url, ClientHttpResponse response) throws IOException
{
    if (logger.isWarnEnabled()) {
        try {
            logger.warn("Async " + method.name() + " request for \"" + url +
                "\" resulted in " + response.getStatusCode() + " (" +
                    response.getStatusText() + "); invoking error handler");
        }
        catch (IOException ex) {
            // ignore
        }
    }
    getErrorHandler().handleError(response);
}

/**
 * Returns a request callback implementation that prepares the request { @code Accept}
 * headers based on the given response type and configured { @link plain
 * #getMessageConverters() message converters}.
 */
protected <T> AsyncRequestCallback acceptHeaderRequestCallback(Class<T> responseType) {
    return new AsyncRequestCallbackAdapter(this.syncTemplate.acceptHeaderRequestCallback(responseType));
}

/**
 * Returns a request callback implementation that writes the given object to the

```

```

* request stream.
*/
protected <T> AsyncRequestCallback httpEntityCallback(HttpEntity<T> requestBody) {
    return new AsyncRequestCallbackAdapter(this.syncTemplate.httpEntityCallback(requestBody));
}

/**
 * Returns a request callback implementation that writes the given object to the
 * request stream.
 */
protected <T> AsyncRequestCallback httpEntityCallback(HttpEntity<T> request, Type responseType) {
    return new AsyncRequestCallbackAdapter(this.syncTemplate.httpEntityCallback(request, responseType));
}

/**
 * Returns a response extractor for { @link ResponseEntity }.
 */
protected <T> ResponseExtractor<ResponseEntity<T>> responseEntityExtractor(Type responseType) {
    return this.syncTemplate.responseEntityExtractor(responseType);
}

/**
 * Returns a response extractor for { @link HttpHeaders }.
 */
protected ResponseExtractor<HttpHeaders> headersExtractor() {
    return this.syncTemplate.headersExtractor();
}

/**
 * Future returned from
 * { @link #doExecute(URI, HttpMethod, AsyncRequestCallback, ResponseExtractor) }
 */
private class ResponseExtractorFuture<T>
    extends ListenableFutureAdapter<T, ClientHttpResponse> {

    private final HttpMethod method;

    private final URI url;

    private final ResponseExtractor<T> responseExtractor;

    public ResponseExtractorFuture(HttpMethod method, URI url,
        ListenableFuture<ClientHttpResponse> clientHttpResponseFuture,
        ResponseExtractor<T> responseExtractor) {
        super(clientHttpResponseFuture);
        this.method = method;
        this.url = url;
        this.responseExtractor = responseExtractor;
    }
}

```



```

}

@Override
protected final T adapt(ClientHttpResponse response) throws ExecutionException {
    try {
        if (!getErrorHandler().hasError(response)) {
            logResponseStatus(this.method, this.url, response);
        }
        else {
            handleResponseError(this.method, this.url, response);
        }
        return convertResponse(response);
    }
    catch (IOException ex) {
        throw new ExecutionException(ex);
    }
    finally {
        if (response != null) {
            response.close();
        }
    }
}

protected T convertResponse(ClientHttpResponse response) throws IOException {
    return responseExtractor != null ? responseExtractor.extractData(response) :
        null;
}

}

/**
 * Adapts a {@link RequestCallback} to the {@link AsyncRequestCallback} interface.
 */
private static class AsyncRequestCallbackAdapter implements AsyncRequestCallback {

    private final RequestCallback adaptee;

    /**
     * Create a new {@code AsyncRequestCallbackAdapter} from the given
     * {@code RequestCallback}.
     * @param requestCallback the callback to base this adapter on
     */
    public AsyncRequestCallbackAdapter(RequestCallback requestCallback) {
        this.adaptee = requestCallback;
    }

    @Override
    public void doWithRequest(final AsyncClientHttpRequest request) throws IOException {

```

```

if (this.adaptee != null) {
    this.adaptee.doWithRequest(new ClientHttpRequest() {
        @Override
        public ClientHttpResponse execute() throws IOException {
            throw new UnsupportedOperationException("execute not supported");
        }
        @Override
        public OutputStream getBody() throws IOException {
            return request.getBody();
        }
        @Override
        public HttpMethod getMethod() {
            return request.getMethod();
        }
        @Override
        public URI getURI() {
            return request.getURI();
        }
        @Override
        public HttpHeaders getHeaders() {
            return request.getHeaders();
        }
    });
}

}

/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.http.client;

import java.io.IOException;
import java.net.HttpURLConnection;

```

```

import java.net.Proxy;
import java.net.URI;
import java.net.URL;
import java.net.URLConnection;

import org.springframework.core.task.AsyncListenableTaskExecutor;
import org.springframework.http.HttpMethod;
import org.springframework.util.Assert;

/**
 * {@link ClientHttpRequestFactory} implementation that uses standard J2SE facilities.
 *
 * @author Arjen Poutsma
 * @author Juergen Hoeller
 * @since 3.0
 * @see java.net.HttpURLConnection
 * @see HttpComponentsClientHttpRequestFactory
 */
public class SimpleClientHttpRequestFactory
    implements ClientHttpRequestFactory, AsyncClientHttpRequestFactory {

    private static final int DEFAULT_CHUNK_SIZE = 4096;

    private Proxy proxy;

    private boolean bufferRequestBody = true;

    private int chunkSize = DEFAULT_CHUNK_SIZE;

    private int connectTimeout = -1;

    private int readTimeout = -1;

    private boolean outputStreaming = true;

    private AsyncListenableTaskExecutor taskExecutor;

    /**
     * Set the {@link Proxy} to use for this request factory.
     */
    public void setProxy(Proxy proxy) {
        this.proxy = proxy;
    }

    /**
     * Indicates whether this request factory should buffer the {@link plain ClientHttpRequest#getBody() request body}

```

```

* internally.
* <p>Default is { @code true}. When sending large amounts of data via POST or PUT, it is recommended
* to change this property to { @code false}, so as not to run out of memory. This will result in a
* { @link ClientHttpRequest} that either streams directly to the underlying { @link HttpURLConnection}
* (if the { @link org.springframework.http.HttpHeaders#getContentLength() Content-Length} is known in
advance),
* or that will use "Chunked transfer encoding" (if the { @code Content-Length} is not known in advance).
* @see #setChunkSize(int)
* @see HttpURLConnection#setFixedLengthStreamingMode(int)
*/
public void setBufferRequestBody(boolean bufferRequestBody) {
    this.bufferRequestBody = bufferRequestBody;
}

/**
* Sets the number of bytes to write in each chunk when not buffering request bodies locally.
* <p>Note that this parameter is only used when { @link #setBufferRequestBody(boolean) bufferRequestBody} is
set
* to { @code false}, and the { @link org.springframework.http.HttpHeaders#getContentLength() Content-Length}
* is not known in advance.
* @see #setBufferRequestBody(boolean)
*/
public void setChunkSize(int chunkSize) {
    this.chunkSize = chunkSize;
}

/**
* Set the underlying URLConnection's connect timeout (in milliseconds).
* A timeout value of 0 specifies an infinite timeout.
* <p>Default is the system's default timeout.
* @see URLConnection#setConnectTimeout(int)
*/
public void setConnectTimeout(int connectTimeout) {
    this.connectTimeout = connectTimeout;
}

/**
* Set the underlying URLConnection's read timeout (in milliseconds).
* A timeout value of 0 specifies an infinite timeout.
* <p>Default is the system's default timeout.
* @see URLConnection#setReadTimeout(int)
*/
public void setReadTimeout(int readTimeout) {
    this.readTimeout = readTimeout;
}

/**
* Set if the underlying URLConnection can be set to 'output streaming' mode. When

```

```

* output streaming is enabled, authentication and redirection cannot be handled
* automatically. If output streaming is disabled the
* { @link HttpURLConnection#setFixedLengthStreamingMode(int)
* setFixedLengthStreamingMode} and
* { @link HttpURLConnection#setChunkedStreamingMode(int) setChunkedStreamingMode}
* methods of the underlying connection will never be called.
* <p>Default is { @code true}.
* @param outputStreaming if output streaming is enabled
*/
public void setOutputStreaming(boolean outputStreaming) {
    this.outputStreaming = outputStreaming;
}

/**
* Sets the task executor for this request factory. Setting this property is required
* for { @linkplain #createAsyncRequest(URI, HttpMethod) creating asynchronous
* request}.
* @param taskExecutor the task executor
*/
public void setTaskExecutor(AsyncListenableTaskExecutor taskExecutor) {
    this.taskExecutor = taskExecutor;
}

@Override
public ClientHttpRequest createRequest(URI uri, HttpMethod httpMethod) throws IOException {
    HttpURLConnection connection = openConnection(uri.toURL(), this.proxy);
    prepareConnection(connection, httpMethod.name());
    if (this.bufferRequestBody) {
        return new SimpleBufferingClientHttpRequest(connection, this.outputStreaming);
    }
    else {
        return new SimpleStreamingClientHttpRequest(connection, this.chunkSize, this.outputStreaming);
    }
}

/**
* { @inheritDoc}
* <p>Setting the { @link #setTaskExecutor(org.springframework.core.task.AsyncListenableTaskExecutor)
taskExecutor} property
* is required before calling this method.
*/
@Override
public AsyncClientHttpRequest createAsyncRequest(URI uri, HttpMethod httpMethod)
    throws IOException {
    Assert.state(this.taskExecutor != null, "Asynchronous execution requires an " +
        "AsyncTaskExecutor to be set");
    HttpURLConnection connection = openConnection(uri.toURL(), this.proxy);
    prepareConnection(connection, httpMethod.name());
}

```

```

if (this.bufferRequestBody) {
    return new SimpleBufferingAsyncClientHttpRequest(connection, this.outputStreaming, this.taskExecutor);
}
else {
    return new SimpleStreamingAsyncClientHttpRequest(connection, this.chunkSize,
        this.outputStreaming, this.taskExecutor);
}
}

/**
 * Opens and returns a connection to the given URL.
 * <p>The default implementation uses the given { @linkplain #setProxy(java.net.Proxy) proxy} -
 * if any - to open a connection.
 * @param url the URL to open a connection to
 * @param proxy the proxy to use, may be { @code null}
 * @return the opened connection
 * @throws IOException in case of I/O errors
 */
protected HttpURLConnection openConnection(URL url, Proxy proxy) throws IOException {
    HttpURLConnection urlConnection = (proxy != null ? url.openConnection(proxy) : url.openConnection());
    Assert.isInstanceOf(HttpURLConnection.class, urlConnection);
    return (HttpURLConnection) urlConnection;
}

/**
 * Template method for preparing the given { @link HttpURLConnection}.
 * <p>The default implementation prepares the connection for input and output, and sets the HTTP method.
 * @param connection the connection to prepare
 * @param httpMethod the HTTP request method ({ @code GET}, { @code POST}, etc.)
 * @throws IOException in case of I/O errors
 */
protected void prepareConnection(HttpURLConnection connection, String httpMethod) throws IOException {
    if (this.connectTimeout >= 0) {
        connection.setConnectTimeout(this.connectTimeout);
    }
    if (this.readTimeout >= 0) {
        connection.setReadTimeout(this.readTimeout);
    }
    connection.setDoInput(true);
    if ("GET".equals(httpMethod)) {
        connection.setInstanceFollowRedirects(true);
    }
    else {
        connection.setInstanceFollowRedirects(false);
    }
    if ("PUT".equals(httpMethod) || "POST".equals(httpMethod) || "PATCH".equals(httpMethod)) {
        connection.setDoOutput(true);
    }
}

```

```

else {
    connection.setDoOutput(false);
}
connection.setRequestMethod(httpMethod);
}

}
/*
 * Copyright 2002-2012 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.http.client;

import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;

import org.springframework.http.HttpHeaders;
import org.springframework.util.StringUtils;

/**
 * {@link ClientHttpResponse} implementation that uses standard J2SE facilities.
 * Obtained via {@link SimpleBufferingClientHttpRequest#execute()} and
 * {@link SimpleStreamingClientHttpRequest#execute()}.
 *
 * @author Arjen Poutsma
 * @since 3.0
 */
final class SimpleClientHttpResponse extends AbstractClientHttpResponse {

    private final HttpURLConnection connection;

    private HttpHeaders headers;

    SimpleClientHttpResponse(HttpURLConnection connection) {

```

```

this.connection = connection;
}

@Override
public int getRawStatusCode() throws IOException {
    return this.connection.getResponseCode();
}

@Override
public String getStatusText() throws IOException {
    return this.connection.getResponseMessage();
}

@Override
public HttpHeaders getHeaders() {
    if (this.headers == null) {
        this.headers = new HttpHeaders();
        // Header field 0 is the status line for most HttpURLConnections, but not on GAE
        String name = this.connection.getHeaderFieldKey(0);
        if (StringUtil.hasLength(name)) {
            this.headers.add(name, this.connection.getHeaderField(0));
        }
        int i = 1;
        while (true) {
            name = this.connection.getHeaderFieldKey(i);
            if (!StringUtil.hasLength(name)) {
                break;
            }
            this.headers.add(name, this.connection.getHeaderField(i));
            i++;
        }
    }
    return this.headers;
}

@Override
public InputStream getBody() throws IOException {
    InputStream errorStream = this.connection.getErrorStream();
    return (errorStream != null ? errorStream : this.connection.getInputStream());
}

@Override
public void close() {
    this.connection.disconnect();
}
}

```



```

/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.http.client;

import java.io.IOException;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.List;
import java.util.Map;

import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.util.StreamUtils;

/**
 * {@link ClientHttpRequest} implementation that uses standard J2SE facilities to execute streaming requests.
 * Created via the {@link SimpleClientHttpRequestFactory}.
 *
 * @author Arjen Poutsma
 * @since 3.0
 * @see SimpleClientHttpRequestFactory#createRequest(java.net.URI, HttpMethod)
 */
final class SimpleStreamingClientHttpRequest extends AbstractClientHttpRequest {

    private final HttpURLConnection connection;

    private final int chunkSize;

    private OutputStream body;

    private final boolean outputStreaming;

```

```

SimpleStreamingClientHttpRequest(HttpURLConnection connection, int chunkSize,
    boolean outputStreaming) {
    this.connection = connection;
    this.chunkSize = chunkSize;
    this.outputStreaming = outputStreaming;
}

@Override
public HttpMethod getMethod() {
    return HttpMethod.valueOf(this.connection.getRequestMethod());
}

@Override
public URI getURI() {
    try {
        return this.connection.getURL().toURI();
    }
    catch (URISyntaxException ex) {
        throw new IllegalStateException("Could not get HttpURLConnection URI: " + ex.getMessage(), ex);
    }
}

@Override
protected OutputStream getBodyInternal(HttpHeaders headers) throws IOException {
    if (this.body == null) {
        if(this.outputStreaming) {
            int contentLength = (int) headers.getContentLength();
            if (contentLength >= 0) {
                this.connection.setFixedLengthStreamingMode(contentLength);
            }
            else {
                this.connection.setChunkedStreamingMode(this.chunkSize);
            }
        }
        writeHeaders(headers);
        this.connection.connect();
        this.body = this.connection.getOutputStream();
    }
    return StreamUtils.nonClosing(this.body);
}

private void writeHeaders(HttpHeaders headers) {
    for (Map.Entry<String, List<String>> entry : headers.entrySet()) {
        String headerName = entry.getKey();
        for (String headerValue : entry.getValue()) {
            this.connection.addRequestProperty(headerName, headerValue);
        }
    }
}

```

```

    }
}

@Override
protected ClientHttpResponse executeInternal(HttpHeaders headers) throws IOException {
    try {
        if (this.body != null) {
            this.body.close();
        }
        else {
            writeHeaders(headers);
            this.connection.connect();
        }
    }
    catch (IOException ex) {
        // ignore
    }
    return new SimpleClientHttpResponse(this.connection);
}

}
/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.http.client;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.List;
import java.util.Map;

import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;

```

```

import org.springframework.util.FileCopyUtils;

/**
 * {@link ClientHttpRequest} implementation that uses standard J2SE facilities to execute buffered requests.
 * Created via the {@link SimpleClientHttpRequestFactory}.
 *
 * @author Arjen Poutsma
 * @since 3.0
 * @see SimpleClientHttpRequestFactory#createRequest(java.net.URI, HttpMethod)
 */
final class SimpleBufferingClientHttpRequest extends AbstractBufferingClientHttpRequest {

    private final HttpURLConnection connection;

    private final boolean outputStreaming;

    SimpleBufferingClientHttpRequest(HttpURLConnection connection, boolean outputStreaming) {
        this.connection = connection;
        this.outputStreaming = outputStreaming;
    }

    @Override
    public HttpMethod getMethod() {
        return HttpMethod.valueOf(this.connection.getRequestMethod());
    }

    @Override
    public URI getURI() {
        try {
            return this.connection.getURL().toURI();
        }
        catch (URISyntaxException ex) {
            throw new IllegalStateException("Could not get HttpURLConnection URI: " + ex.getMessage(), ex);
        }
    }

    @Override
    protected ClientHttpResponse executeInternal(HttpHeaders headers, byte[] bufferedOutput) throws IOException {
        for (Map.Entry<String, List<String>> entry : headers.entrySet()) {
            String headerName = entry.getKey();
            for (String headerValue : entry.getValue()) {
                this.connection.addRequestProperty(headerName, headerValue);
            }
        }

        if (this.connection.getDoOutput() && this.outputStreaming) {

```

```

        this.connection.setFixedLengthStreamingMode(bufferedOutput.length);
    }
    this.connection.connect();
    if (this.connection.getDoOutput()) {
        FileCopyUtils.copy(bufferedOutput, this.connection.getOutputStream());
    }

    return new SimpleClientHttpResponse(this.connection);
}

}
/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.http.client;

import java.io.IOException;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.List;
import java.util.Map;
import java.util.concurrent.Callable;

import org.springframework.core.task.AsyncListenableTaskExecutor;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.util.StreamUtils;
import org.springframework.util.concurrent.ListenableFuture;

/**
 * {@link org.springframework.http.client.ClientHttpRequest} implementation that uses
 * standard J2SE facilities to execute streaming requests. Created via the {@link
 * org.springframework.http.client.SimpleClientHttpRequestFactory}.

```

```

*
* @author Arjen Poutsma
* @see org.springframework.http.client.SimpleClientHttpRequestFactory#createRequest(java.net.URI,
*     org.springframework.http.HttpMethod)
* @since 3.0
*/
final class SimpleStreamingAsyncClientHttpRequest extends AbstractAsyncClientHttpRequest {

    private final HttpURLConnection connection;

    private final int chunkSize;

    private OutputStream body;

    private final boolean outputStreaming;

    private final AsyncListenableTaskExecutor taskExecutor;

    SimpleStreamingAsyncClientHttpRequest(HttpURLConnection connection, int chunkSize,
        boolean outputStreaming, AsyncListenableTaskExecutor taskExecutor) {
        this.connection = connection;
        this.chunkSize = chunkSize;
        this.outputStreaming = outputStreaming;
        this.taskExecutor = taskExecutor;
    }

    @Override
    public HttpMethod getMethod() {
        return HttpMethod.valueOf(this.connection.getRequestMethod());
    }

    @Override
    public URI getURI() {
        try {
            return this.connection.getURL().toURI();
        }
        catch (URISyntaxException ex) {
            throw new IllegalStateException(
                "Could not get HttpURLConnection URI: " + ex.getMessage(), ex);
        }
    }

    @Override
    protected OutputStream getBodyInternal(HttpHeaders headers) throws IOException {
        if (this.body == null) {
            if (this.outputStreaming) {
                int contentLength = (int) headers.getContentLength();
                if (contentLength >= 0) {

```

```

        this.connection.setFixedLengthStreamingMode(contentLength);
    }
    else {
        this.connection.setChunkedStreamingMode(this.chunkSize);
    }
}
writeHeaders(headers);
this.connection.connect();
this.body = this.connection.getOutputStream();
}
return StreamUtils.nonClosing(this.body);
}

private void writeHeaders(HttpHeaders headers) {
    for (Map.Entry<String, List<String>> entry : headers.entrySet()) {
        String headerName = entry.getKey();
        for (String headerValue : entry.getValue()) {
            this.connection.addRequestProperty(headerName, headerValue);
        }
    }
}

@Override
protected ListenableFuture<ClientHttpResponse> executeInternal(final HttpHeaders headers)
    throws IOException {
    return taskExecutor.submitListenable(new Callable<ClientHttpResponse>() {
        @Override
        public ClientHttpResponse call() throws Exception {
            try {
                if (body != null) {
                    body.close();
                }
                else {
                    writeHeaders(headers);
                    connection.connect();
                }
            }
            catch (IOException ex) {
                // ignore
            }
            return new SimpleClientHttpResponse(connection);
        }
    });
}
}
}

```

```

/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.http.client;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URI;
import java.net.URISyntaxException;
import java.util.List;
import java.util.Map;
import java.util.concurrent.Callable;

import org.springframework.core.task.AsyncListenableTaskExecutor;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.util.FileCopyUtils;
import org.springframework.util.concurrent.ListenableFuture;

/**
 * {@link org.springframework.http.client.ClientHttpRequest} implementation that uses
 * standard J2SE facilities to execute buffered requests. Created via the
 * {@link org.springframework.http.client.SimpleClientHttpRequestFactory}.
 *
 * @author Arjen Poutsma
 * @since 3.0
 * @see org.springframework.http.client.SimpleClientHttpRequestFactory#createRequest(java.net.URI,
 * org.springframework.http.HttpMethod)
 */
final class SimpleBufferingAsyncClientHttpRequest extends AbstractBufferingAsyncClientHttpRequest {

    private final HttpURLConnection connection;

    private final boolean outputStreaming;

```



```

private final AsyncListenableTaskExecutor taskExecutor;

SimpleBufferingAsyncClientHttpRequest(HttpURLConnection connection,
    boolean outputStreaming, AsyncListenableTaskExecutor taskExecutor) {
    this.connection = connection;
    this.outputStreaming = outputStreaming;
    this.taskExecutor = taskExecutor;
}

@Override
public HttpMethod getMethod() {
    return HttpMethod.valueOf(this.connection.getRequestMethod());
}

@Override
public URI getURI() {
    try {
        return this.connection.getURL().toURI();
    }
    catch (URISyntaxException ex) {
        throw new IllegalStateException("Could not get HttpURLConnection URI: " + ex.getMessage(), ex);
    }
}

@Override
protected ListenableFuture<ClientHttpResponse> executeInternal(
    final HttpHeaders headers, final byte[] bufferedOutput) throws IOException {
    return taskExecutor.submitListenable(new Callable<ClientHttpResponse>() {
        @Override
        public ClientHttpResponse call() throws Exception {
            for (Map.Entry<String, List<String>> entry : headers.entrySet()) {
                String headerName = entry.getKey();
                for (String headerValue : entry.getValue()) {
                    connection.addRequestProperty(headerName, headerValue);
                }
            }

            if (connection.getDoOutput() && outputStreaming) {
                connection.setFixedLengthStreamingMode(bufferedOutput.length);
            }

            connection.connect();
            if (connection.getDoOutput()) {
                FileCopyUtils.copy(bufferedOutput, connection.getOutputStream());
            }
            return new SimpleClientHttpResponse(connection);
        }
    });
}

```

```

}

}
/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.remoting.httpinvoker;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import java.util.Locale;
import java.util.zip.GZIPInputStream;

import org.springframework.context.i18n.LocaleContext;
import org.springframework.context.i18n.LocaleContextHolder;
import org.springframework.remoting.support.RemoteInvocationResult;
import org.springframework.util.StringUtils;

/**
 * HttpInvokerRequestExecutor implementation that uses standard J2SE facilities
 * to execute POST requests, without support for HTTP authentication or
 * advanced configuration options.
 *
 * <p>Designed for easy subclassing, customizing specific template methods.
 * However, consider CommonsHttpInvokerRequestExecutor for more sophisticated
 * needs: The J2SE HttpURLConnection is rather limited in its capabilities.
 *
 * @author Juergen Hoeller
 * @since 1.1
 * @see java.net.HttpURLConnection
 */

```

```

public class SimpleHttpInvokerRequestExecutor extends AbstractHttpInvokerRequestExecutor {

    private int connectTimeout = -1;

    private int readTimeout = -1;

    /**
     * Set the underlying URLConnection's connect timeout (in milliseconds).
     * A timeout value of 0 specifies an infinite timeout.
     * <p>Default is the system's default timeout.
     * @see URLConnection#setConnectTimeout(int)
     */
    public void setConnectTimeout(int connectTimeout) {
        this.connectTimeout = connectTimeout;
    }

    /**
     * Set the underlying URLConnection's read timeout (in milliseconds).
     * A timeout value of 0 specifies an infinite timeout.
     * <p>Default is the system's default timeout.
     * @see URLConnection#setReadTimeout(int)
     */
    public void setReadTimeout(int readTimeout) {
        this.readTimeout = readTimeout;
    }

    /**
     * Execute the given request through a standard J2SE HttpURLConnection.
     * <p>This method implements the basic processing workflow:
     * The actual work happens in this class's template methods.
     * @see #openConnection
     * @see #prepareConnection
     * @see #writeRequestBody
     * @see #validateResponse
     * @see #readResponseBody
     */
    @Override
    protected RemoteInvocationResult doExecuteRequest(
        HttpInvokerClientConfiguration config, ByteArrayOutputStream baos)
        throws IOException, ClassNotFoundException {

        HttpURLConnection con = openConnection(config);
        prepareConnection(con, baos.size());
        writeRequestBody(config, con, baos);
        validateResponse(config, con);
        InputStream responseBody = readResponseBody(config, con);
    }
}

```

```

return readRemoteInvocationResult(responseBody, config.getCodebaseUrl());
}

/**
 * Open an HttpURLConnection for the given remote invocation request.
 * @param config the HTTP invoker configuration that specifies the
 * target service
 * @return the HttpURLConnection for the given request
 * @throws IOException if thrown by I/O methods
 * @see java.net.URL#openConnection()
 */
protected HttpURLConnection openConnection(HttpInvokerClientConfiguration config) throws IOException {
    URLConnection con = new URL(config.getServiceUrl()).openConnection();
    if (!(con instanceof HttpURLConnection)) {
        throw new IOException("Service URL [" + config.getServiceUrl() + "] is not an HTTP URL");
    }
    return (HttpURLConnection) con;
}

/**
 * Prepare the given HTTP connection.
 * <p>The default implementation specifies POST as method,
 * "application/x-java-serialized-object" as "Content-Type" header,
 * and the given content length as "Content-Length" header.
 * @param connection the HTTP connection to prepare
 * @param contentLength the length of the content to send
 * @throws IOException if thrown by HttpURLConnection methods
 * @see java.net.HttpURLConnection#setRequestMethod
 * @see java.net.HttpURLConnection#setRequestProperty
 */
protected void prepareConnection(HttpURLConnection connection, int contentLength) throws IOException {
    if (this.connectTimeout >= 0) {
        connection.setConnectTimeout(this.connectTimeout);
    }
    if (this.readTimeout >= 0) {
        connection.setReadTimeout(this.readTimeout);
    }
    connection.setDoOutput(true);
    connection.setRequestMethod(HTTP_METHOD_POST);
    connection.setRequestProperty(HTTP_HEADER_CONTENT_TYPE, getContentType());
    connection.setRequestProperty(HTTP_HEADER_CONTENT_LENGTH, Integer.toString(contentLength));

    LocaleContext localeContext = LocaleContextHolder.getLocaleContext();
    if (localeContext != null) {
        Locale locale = localeContext.getLocale();
        if (locale != null) {
            connection.setRequestProperty(HTTP_HEADER_ACCEPT_LANGUAGE, StringUtils.toLanguageTag(locale));
        }
    }
}

```

```

    }
    }
    if (isAcceptGzipEncoding()) {
        connection.setRequestProperty(HTTP_HEADER_ACCEPT_ENCODING, ENCODING_GZIP);
    }
}

/**
 * Set the given serialized remote invocation as request body.
 * <p>The default implementation simply write the serialized invocation to the
 * HttpURLConnection's OutputStream. This can be overridden, for example, to write
 * a specific encoding and potentially set appropriate HTTP request headers.
 * @param config the HTTP invoker configuration that specifies the target service
 * @param con the HttpURLConnection to write the request body to
 * @param baos the ByteArrayOutputStream that contains the serialized
 * RemoteInvocation object
 * @throws IOException if thrown by I/O methods
 * @see java.net.HttpURLConnection#getOutputStream()
 * @see java.net.HttpURLConnection#setRequestProperty
 */
protected void writeRequestBody(
    HttpInvokerClientConfiguration config, HttpURLConnection con, ByteArrayOutputStream baos)
    throws IOException {

    baos.writeTo(con.getOutputStream());
}

/**
 * Validate the given response as contained in the HttpURLConnection object,
 * throwing an exception if it does not correspond to a successful HTTP response.
 * <p>Default implementation rejects any HTTP status code beyond 2xx, to avoid
 * parsing the response body and trying to deserialize from a corrupted stream.
 * @param config the HTTP invoker configuration that specifies the target service
 * @param con the HttpURLConnection to validate
 * @throws IOException if validation failed
 * @see java.net.HttpURLConnection#getResponseCode()
 */
protected void validateResponse(HttpInvokerClientConfiguration config, HttpURLConnection con)
    throws IOException {

    if (con.getResponseCode() >= 300) {
        throw new IOException(
            "Did not receive successful HTTP response: status code = " + con.getResponseCode() +
            ", status message = [" + con.getResponseMessage() + "]);
    }
}

/**

```

```

* Extract the response body from the given executed remote invocation
* request.
* <p>The default implementation simply reads the serialized invocation
* from the HttpURLConnection's InputStream. If the response is recognized
* as GZIP response, the InputStream will get wrapped in a GZIPInputStream.
* @param config the HTTP invoker configuration that specifies the target service
* @param con the HttpURLConnection to read the response body from
* @return an InputStream for the response body
* @throws IOException if thrown by I/O methods
* @see #isGzipResponse
* @see java.util.zip.GZIPInputStream
* @see java.net.HttpURLConnection#getInputStream()
* @see java.net.HttpURLConnection#getHeaderField(int)
* @see java.net.HttpURLConnection#getHeaderFieldKey(int)
*/
protected InputStream readResponseBody(HttpInvokerClientConfiguration config, HttpURLConnection con)
    throws IOException {

    if (isGzipResponse(con)) {
        // GZIP response found - need to unzip.
        return new GZIPInputStream(con.getInputStream());
    }
    else {
        // Plain response found.
        return con.getInputStream();
    }
}

/**
* Determine whether the given response is a GZIP response.
* <p>Default implementation checks whether the HTTP "Content-Encoding"
* header contains "gzip" (in any casing).
* @param con the HttpURLConnection to check
*/
protected boolean isGzipResponse(HttpURLConnection con) {
    String encodingHeader = con.getHeaderField(HTTP_HEADER_CONTENT_ENCODING);
    return (encodingHeader != null && encodingHeader.toLowerCase().contains(ENCODING_GZIP));
}

}

/*
* Copyright 2002-2012 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0

```

```

*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

```

```
package org.springframework.remoting.httpinvoker;
```

```
import java.io.IOException;
import java.io.InputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.OutputStream;
```

```
import com.sun.net.httpserver.HttpExchange;
import com.sun.net.httpserver.HttpHandler;
```

```
import org.springframework.remoting.rmi.RemoteInvocationSerializingExporter;
import org.springframework.remoting.support.RemoteInvocation;
import org.springframework.remoting.support.RemoteInvocationResult;
```

```
/**
 * HTTP request handler that exports the specified service bean as
 * HTTP invoker service endpoint, accessible via an HTTP invoker proxy.
 * Designed for Sun's JRE 1.6 HTTP server, implementing the
 * { @link com.sun.net.httpserver.HttpHandler } interface.
 *
 * <p>Deserializes remote invocation objects and serializes remote invocation
 * result objects. Uses Java serialization just like RMI, but provides the
 * same ease of setup as Caucho's HTTP-based Hessian and Burlap protocols.
 *
 * <p><b>HTTP invoker is the recommended protocol for Java-to-Java remoting.</b>
 * It is more powerful and more extensible than Hessian and Burlap, at the
 * expense of being tied to Java. Nevertheless, it is as easy to set up as
 * Hessian and Burlap, which is its main advantage compared to RMI.
 *
 * @author Juergen Hoeller
 * @since 2.5.1
 * @see org.springframework.remoting.httpinvoker.HttpInvokerClientInterceptor
 * @see org.springframework.remoting.httpinvoker.HttpInvokerProxyFactoryBean
 * @see org.springframework.remoting.caucho.SimpleHessianServiceExporter
 * @see org.springframework.remoting.caucho.SimpleBurlapServiceExporter
 */

```

```
public class SimpleHttpInvokerServiceExporter extends RemoteInvocationSerializingExporter
implements HttpHandler {
```

```

/**
 * Reads a remote invocation from the request, executes it,
 * and writes the remote invocation result to the response.
 * @see #readRemoteInvocation(com.sun.net.httpserver.HttpExchange)
 * @see #invokeAndCreateResult(org.springframework.remoting.support.RemoteInvocation, Object)
 * @see #writeRemoteInvocationResult(com.sun.net.httpserver.HttpExchange,
org.springframework.remoting.support.RemoteInvocationResult)
 */
@Override
public void handle(HttpExchange exchange) throws IOException {
    try {
        RemoteInvocation invocation = readRemoteInvocation(exchange);
        RemoteInvocationResult result = invokeAndCreateResult(invocation, getProxy());
        writeRemoteInvocationResult(exchange, result);
        exchange.close();
    }
    catch (ClassNotFoundException ex) {
        exchange.sendResponseHeaders(500, -1);
        logger.error("Class not found during deserialization", ex);
    }
}

/**
 * Read a RemoteInvocation from the given HTTP request.
 * <p>Delegates to
 * { @link #readRemoteInvocation(com.sun.net.httpserver.HttpExchange, java.io.InputStream) }
 * with the
 * { @link com.sun.net.httpserver.HttpExchange#getRequestBody() } request's input stream}.
 * @param exchange current HTTP request/response
 * @return the RemoteInvocation object
 * @throws java.io.IOException in case of I/O failure
 * @throws ClassNotFoundException if thrown by deserialization
 */
protected RemoteInvocation readRemoteInvocation(HttpExchange exchange)
    throws IOException, ClassNotFoundException {

    return readRemoteInvocation(exchange, exchange.getRequestBody());
}

/**
 * Deserialize a RemoteInvocation object from the given InputStream.
 * <p>Gives { @link #decorateInputStream } a chance to decorate the stream
 * first (for example, for custom encryption or compression). Creates a
 * { @link org.springframework.remoting.rmi.CodebaseAwareObjectInputStream }
 * and calls { @link #doReadRemoteInvocation } to actually read the object.
 * <p>Can be overridden for custom serialization of the invocation.
 * @param exchange current HTTP request/response
 * @param is the InputStream to read from

```



```

* @return the RemoteInvocation object
* @throws java.io.IOException in case of I/O failure
* @throws ClassNotFoundException if thrown during deserialization
*/
protected RemoteInvocation readRemoteInvocation(HttpExchange exchange, InputStream is)
    throws IOException, ClassNotFoundException {

    ObjectInputStream ois = createObjectInputStream(decorateInputStream(exchange, is));
    return doReadRemoteInvocation(ois);
}

/**
* Return the InputStream to use for reading remote invocations,
* potentially decorating the given original InputStream.
* <p>The default implementation returns the given stream as-is.
* Can be overridden, for example, for custom encryption or compression.
* @param exchange current HTTP request/response
* @param is the original InputStream
* @return the potentially decorated InputStream
* @throws java.io.IOException in case of I/O failure
*/
protected InputStream decorateInputStream(HttpExchange exchange, InputStream is) throws IOException {
    return is;
}

/**
* Write the given RemoteInvocationResult to the given HTTP response.
* @param exchange current HTTP request/response
* @param result the RemoteInvocationResult object
* @throws java.io.IOException in case of I/O failure
*/
protected void writeRemoteInvocationResult(HttpExchange exchange, RemoteInvocationResult result)
    throws IOException {

    exchange.getResponseHeaders().set("Content-Type", getContentType());
    exchange.sendResponseHeaders(200, 0);
    writeRemoteInvocationResult(exchange, result, exchange.getResponseBody());
}

/**
* Serialize the given RemoteInvocation to the given OutputStream.
* <p>The default implementation gives { @link #decorateOutputStream } a chance
* to decorate the stream first (for example, for custom encryption or compression).
* Creates an { @link java.io.ObjectOutputStream } for the final stream and calls
* { @link #doWriteRemoteInvocationResult } to actually write the object.
* <p>Can be overridden for custom serialization of the invocation.
* @param exchange current HTTP request/response
* @param result the RemoteInvocationResult object

```

```

* @param os the OutputStream to write to
* @throws java.io.IOException in case of I/O failure
* @see #decorateOutputStream
* @see #doWriteRemoteInvocationResult
*/
protected void writeRemoteInvocationResult(
    HttpExchange exchange, RemoteInvocationResult result, OutputStream os) throws IOException {

    ObjectOutputStream oos = createObjectOutputStream(decorateOutputStream(exchange, os));
    doWriteRemoteInvocationResult(result, oos);
    oos.flush();
}

/**
* Return the OutputStream to use for writing remote invocation results,
* potentially decorating the given original OutputStream.
* <p>The default implementation returns the given stream as-is.
* Can be overridden, for example, for custom encryption or compression.
* @param exchange current HTTP request/response
* @param os the original OutputStream
* @return the potentially decorated OutputStream
* @throws java.io.IOException in case of I/O failure
*/
protected OutputStream decorateOutputStream(HttpExchange exchange, OutputStream os) throws IOException {
    return os;
}

}

/*
* Copyright 2002-2013 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package org.springframework.remoting.caucho;

import java.io.ByteArrayOutputStream;
import java.io.IOException;

```

```

import com.sun.net.httpserver.HttpExchange;
import com.sun.net.httpserver.HttpHandler;

import org.springframework.util.FileCopyUtils;

/**
 * HTTP request handler that exports the specified service bean as
 * Burlap service endpoint, accessible via a Burlap proxy.
 * Designed for Sun's JRE 1.6 HTTP server, implementing the
 * { @link com.sun.net.httpserver.HttpHandler } interface.
 *
 * <p>Burlap is a slim, XML-based RPC protocol.
 * For information on Burlap, see the
 * <a href="http://www.caucho.com/burlap">Burlap website</a>.
 * This exporter requires Burlap 3.x.
 *
 * <p>Note: Burlap services exported with this class can be accessed by
 * any Burlap client, as there isn't any special handling involved.
 *
 * @author Juergen Hoeller
 * @since 2.5.1
 * @see org.springframework.remoting.caucho.BurlapClientInterceptor
 * @see org.springframework.remoting.caucho.BurlapProxyFactoryBean
 * @see SimpleHessianServiceExporter
 * @see org.springframework.remoting.httpinvoker.SimpleHttpInvokerServiceExporter
 * @deprecated as of Spring 4.0, since Burlap hasn't evolved in years
 * and is effectively retired (in contrast to its sibling Hessian)
 */
@Deprecated
public class SimpleBurlapServiceExporter extends BurlapExporter implements HttpHandler {

    /**
     * Processes the incoming Burlap request and creates a Burlap response.
     */
    @Override
    public void handle(HttpExchange exchange) throws IOException {
        if (!"POST".equals(exchange.getRequestMethod())) {
            exchange.getResponseHeaders().set("Allow", "POST");
            exchange.sendResponseHeaders(405, -1);
            return;
        }

        ByteArrayOutputStream output = new ByteArrayOutputStream(1024);
        try {
            invoke(exchange.getRequestBody(), output);
        }
        catch (Throwable ex) {

```

```

    exchange.sendResponseHeaders(500, -1);
    logger.error("Burlap skeleton invocation failed", ex);
}

exchange.sendResponseHeaders(200, output.size());
FileCopyUtils.copy(output.toByteArray(), exchange.getResponseBody());
}

}
/*
 * Copyright 2002-2013 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.remoting.caucho;

import java.io.ByteArrayOutputStream;
import java.io.IOException;

import com.sun.net.httpserver.HttpExchange;
import com.sun.net.httpserver.HttpHandler;

import org.springframework.util.FileCopyUtils;

/**
 * HTTP request handler that exports the specified service bean as
 * Hessian service endpoint, accessible via a Hessian proxy.
 * Designed for Sun's JRE 1.6 HTTP server, implementing the
 * {@link com.sun.net.httpserver.HttpHandler} interface.
 *
 * <p>Hessian is a slim, binary RPC protocol.
 * For information on Hessian, see the
 * <a href="http://www.caucho.com/hessian">Hessian website</a>.
 * <b>Note: As of Spring 4.0, this exporter requires Hessian 4.0 or above.</b>
 *
 * <p>Hessian services exported with this class can be accessed by
 * any Hessian client, as there isn't any special handling involved.

```

```

*
* @author Juergen Hoeller
* @since 2.5.1
* @see org.springframework.remoting.caucho.HessianClientInterceptor
* @see org.springframework.remoting.caucho.HessianProxyFactoryBean
* @see org.springframework.remoting.httpinvoker.SimpleHttpInvokerServiceExporter
*/
public class SimpleHessianServiceExporter extends HessianExporter implements HttpHandler {

    /**
     * Processes the incoming Hessian request and creates a Hessian response.
     */
    @Override
    public void handle(HttpExchange exchange) throws IOException {
        if (!"POST".equals(exchange.getRequestMethod())) {
            exchange.getResponseHeaders().set("Allow", "POST");
            exchange.sendResponseHeaders(405, -1);
            return;
        }

        ByteArrayOutputStream output = new ByteArrayOutputStream(1024);
        try {
            invoke(exchange.getRequestBody(), output);
        }
        catch (Throwable ex) {
            exchange.sendResponseHeaders(500, -1);
            logger.error("Hessian skeleton invocation failed", ex);
            return;
        }

        exchange.getResponseHeaders().set("Content-Type", CONTENT_TYPE_HESSIAN);
        exchange.sendResponseHeaders(200, output.size());
        FileCopyUtils.copy(output.toByteArray(), exchange.getResponseBody());
    }

    /**
     * Copyright 2002-2009 the original author or authors.
     *
     * Licensed under the Apache License, Version 2.0 (the "License");
     * you may not use this file except in compliance with the License.
     * You may obtain a copy of the License at
     *
     * http://www.apache.org/licenses/LICENSE-2.0
     *
     * Unless required by applicable law or agreed to in writing, software
     * distributed under the License is distributed on an "AS IS" BASIS,
     * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```

\* See the License for the specific language governing permissions and  
\* limitations under the License.  
\*/

```
package org.springframework.remoting.jaxws;
```

```
import java.net.InetSocketAddress;  
import java.util.List;  
import javax.jws.WebService;  
import javax.xml.ws.Endpoint;  
import javax.xml.ws.WebServiceProvider;
```

```
import com.sun.net.httpserver.Authenticator;  
import com.sun.net.httpserver.Filter;  
import com.sun.net.httpserver.HttpContext;  
import com.sun.net.httpserver.HttpServer;  
import org.apache.commons.logging.Log;  
import org.apache.commons.logging.LogFactory;
```

```
/**
```

```
* Simple exporter for JAX-WS services, autodetecting annotated service beans  
* (through the JAX-WS { @link javax.jws.WebService } annotation) and exporting  
* them through the HTTP server included in Sun's JDK 1.6. The full address  
* for each service will consist of the server's base address with the  
* service name appended (e.g. "http://localhost:8080/OrderService").  
*  
* <p>Note that this exporter will only work on Sun's JDK 1.6 or higher, as well  
* as on JDKs that ship Sun's entire class library as included in the Sun JDK.  
* For a portable JAX-WS exporter, have a look at { @link SimpleJaxWsServiceExporter }.  
*  
* @author Juergen Hoeller  
* @since 2.5.5  
* @see javax.jws.WebService  
* @see javax.xml.ws.Endpoint#publish(Object)  
* @see SimpleJaxWsServiceExporter  
*/
```

```
public class SimpleHttpServerJaxWsServiceExporter extends AbstractJaxWsServiceExporter {
```

```
    protected final Log logger = LogFactory.getLog(getClass());
```

```
    private HttpServer server;
```

```
    private int port = 8080;
```

```
    private String hostname;
```

```
    private int backlog = -1;
```

```

private int shutdownDelay = 0;

private String basePath = "/";

private List<Filter> filters;

private Authenticator authenticator;

private boolean localServer = false;

/**
 * Specify an existing HTTP server to register the web service contexts
 * with. This will typically be a server managed by the general Spring
 * { @link org.springframework.remoting.support.SimpleHttpServerFactoryBean }.
 * <p>Alternatively, configure a local HTTP server through the
 * { @link #setPort "port"}, { @link #setHostname "hostname"} and
 * { @link #setBacklog "backlog"} properties (or rely on the defaults there).
 */
public void setServer(HttpServer server) {
    this.server = server;
}

/**
 * Specify the HTTP server's port. Default is 8080.
 * <p>Only applicable for a locally configured HTTP server.
 * Ignored when the { @link #setServer "server"} property has been specified.
 */
public void setPort(int port) {
    this.port = port;
}

/**
 * Specify the HTTP server's hostname to bind to. Default is localhost;
 * can be overridden with a specific network address to bind to.
 * <p>Only applicable for a locally configured HTTP server.
 * Ignored when the { @link #setServer "server"} property has been specified.
 */
public void setHostname(String hostname) {
    this.hostname = hostname;
}

/**
 * Specify the HTTP server's TCP backlog. Default is -1,
 * indicating the system's default value.
 * <p>Only applicable for a locally configured HTTP server.
 * Ignored when the { @link #setServer "server"} property has been specified.
 */

```

```

public void setBacklog(int backlog) {
    this.backlog = backlog;
}

/**
 * Specify the number of seconds to wait until HTTP exchanges have
 * completed when shutting down the HTTP server. Default is 0.
 * <p>Only applicable for a locally configured HTTP server.
 * Ignored when the { @link #setServer "server" } property has been specified.
 */
public void setShutdownDelay(int shutdownDelay) {
    this.shutdownDelay = shutdownDelay;
}

/**
 * Set the base path for context publication. Default is "/".
 * <p>For each context publication path, the service name will be
 * appended to this base address. E.g. service name "OrderService"
 * -> "/OrderService".
 * @see javax.xml.ws.Endpoint#publish(Object)
 * @see javax.jws.WebService#serviceName()
 */
public void setBasePath(String basePath) {
    this.basePath = basePath;
}

/**
 * Register common { @link com.sun.net.httpserver.Filter Filters } to be
 * applied to all detected { @link javax.jws.WebService } annotated beans.
 */
public void setFilters(List<Filter> filters) {
    this.filters = filters;
}

/**
 * Register a common { @link com.sun.net.httpserver.Authenticator } to be
 * applied to all detected { @link javax.jws.WebService } annotated beans.
 */
public void setAuthenticator(Authenticator authenticator) {
    this.authenticator = authenticator;
}

@Override
public void afterPropertiesSet() throws Exception {
    if (this.server == null) {
        InetAddress address = (this.hostname != null ?
            new InetAddress(this.hostname, this.port) : new InetAddress(this.port));
    }
}

```



```

this.server = HttpServer.create(address, this.backlog);
if (this.logger.isInfoEnabled()) {
    this.logger.info("Starting HttpServer at address " + address);
}
this.server.start();
this.localServer = true;
}
super.afterPropertiesSet();
}

@Override
protected void publishEndpoint(Endpoint endpoint, WebService annotation) {
    endpoint.publish(buildHttpContext(endpoint, annotation.serviceName()));
}

@Override
protected void publishEndpoint(Endpoint endpoint, WebServiceProvider annotation) {
    endpoint.publish(buildHttpContext(endpoint, annotation.serviceName()));
}

/**
 * Build the HttpContext for the given endpoint.
 * @param endpoint the JAX-WS Provider Endpoint object
 * @param serviceName the given service name
 * @return the fully populated HttpContext
 */
protected HttpContext buildHttpContext(Endpoint endpoint, String serviceName) {
    String fullPath = calculateEndpointPath(endpoint, serviceName);
    HttpContext httpContext = this.server.createContext(fullPath);
    if (this.filters != null) {
        httpContext.getFilters().addAll(this.filters);
    }
    if (this.authenticator != null) {
        httpContext.setAuthenticator(this.authenticator);
    }
    return httpContext;
}

/**
 * Calculate the full endpoint path for the given endpoint.
 * @param endpoint the JAX-WS Provider Endpoint object
 * @param serviceName the given service name
 * @return the full endpoint path
 */
protected String calculateEndpointPath(Endpoint endpoint, String serviceName) {
    return this.basePath + serviceName;
}

```

```

@Override
public void destroy() {
    super.destroy();
    if (this.localServer) {
        logger.info("Stopping HttpServer");
        this.server.stop(this.shutdownDelay);
    }
}

}
/*
 * Copyright 2002-2009 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.springframework.remoting.jaxws;

import javax.jws.WebService;
import javax.xml.ws.Endpoint;
import javax.xml.ws.WebServiceProvider;

/**
 * Simple exporter for JAX-WS services, autodetecting annotated service beans
 * (through the JAX-WS {@link javax.jws.WebService} annotation) and exporting
 * them with a configured base address (by default "http://localhost:8080/")
 * using the JAX-WS provider's built-in publication support. The full address
 * for each service will consist of the base address with the service name
 * appended (e.g. "http://localhost:8080/OrderService").
 *
 * <p>Note that this exporter will only work if the JAX-WS runtime actually
 * supports publishing with an address argument, i.e. if the JAX-WS runtime
 * ships an internal HTTP server. This is the case with the JAX-WS runtime
 * that's included in Sun's JDK 1.6 but not with the standalone JAX-WS 2.1 RI.
 *
 * <p>For explicit configuration of JAX-WS endpoints with Sun's JDK 1.6
 * HTTP server, consider using {@link SimpleHttpServerJaxWsServiceExporter}!

```

```

*
* @author Juergen Hoeller
* @since 2.5
* @see javax.jws.WebService
* @see javax.xml.ws.Endpoint#publish(String)
* @see SimpleHttpServerJaxWsServiceExporter
*/
public class SimpleJaxWsServiceExporter extends AbstractJaxWsServiceExporter {

    public static final String DEFAULT_BASE_ADDRESS = "http://localhost:8080/";

    private String baseAddress = DEFAULT_BASE_ADDRESS;

    /**
     * Set the base address for exported services.
     * Default is "http://localhost:8080/".
     * <p>For each actual publication address, the service name will be
     * appended to this base address. E.g. service name "OrderService"
     * -> "http://localhost:8080/OrderService".
     * @see javax.xml.ws.Endpoint#publish(String)
     * @see javax.jws.WebService#serviceName()
     */
    public void setBaseAddress(String baseAddress) {
        this.baseAddress = baseAddress;
    }

    @Override
    protected void publishEndpoint(Endpoint endpoint, WebService annotation) {
        endpoint.publish(calculateEndpointAddress(endpoint, annotation.serviceName()));
    }

    @Override
    protected void publishEndpoint(Endpoint endpoint, WebServiceProvider annotation) {
        endpoint.publish(calculateEndpointAddress(endpoint, annotation.serviceName()));
    }

    /**
     * Calculate the full endpoint address for the given endpoint.
     * @param endpoint the JAX-WS Provider Endpoint object
     * @param serviceName the given service name
     * @return the full endpoint address
     */
    protected String calculateEndpointAddress(Endpoint endpoint, String serviceName) {
        String fullAddress = this.baseAddress + serviceName;
        if (endpoint.getClass().getName().startsWith("weblogic.")) {
            // Workaround for WebLogic 10.3

```



"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability











1. A link or URL to the original W3C document.
2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright ) [\$date-of-document] World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231>"
3. If it exists, the STATUS of the W3C document.

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of W3C documents is granted pursuant to this license. However, if additional requirements (documented in the Copyright FAQ) are satisfied, the right to create modifications or derivatives is sometimes granted by the W3C to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

-----

This formulation of W3C's notice and license became active on December 31 2002. This version removes the copyright ownership notice such that this license can be used with materials other than those owned by the W3C, moves information on style sheets, DTDs, and schemas to the Copyright FAQ, reflects that ERCIM is now a host of the W3C, includes references to this specific dated version of the license, and removes the ambiguous grant of "use". See the older formulation for the policy prior to this date. Please see our Copyright FAQ for common questions about using materials from our site, such as the translating or annotating specifications. Other questions about this notice can be directed















- software copyright (c) 1999-2002, Lotus Development Corporation.,  
<http://www.lotus.com>.
- software copyright (c) 2001-2002, Sun Microsystems.,  
<http://www.sun.com>.
- software copyright (c) 2003, IBM Corporation.,  
<http://www.ibm.com>.

---

The binary distribution package (ie. jars, samples and documentation) of this product includes software developed by the following:

- The Apache Software Foundation
  - Xerces Java - see LICENSE.txt
  - JAXP 1.3 APIs - see LICENSE.txt
  - Bytecode Engineering Library - see LICENSE.txt
  - Regular Expression - see LICENSE.txt
- Scott Hudson, Frank Flannery, C. Scott Ananian
  - CUP Parser Generator runtime (javacup\runtime) - see LICENSE.txt

---

The source distribution package (ie. all source and tools required to build Xalan Java) of this product includes software developed by the following:

- The Apache Software Foundation
  - Xerces Java - see LICENSE.txt
  - JAXP 1.3 APIs - see LICENSE.txt
  - Bytecode Engineering Library - see LICENSE.txt
  - Regular Expression - see LICENSE.txt
  - Ant - see LICENSE.txt
  - Stylebook doc tool - see LICENSE.txt
- Elliot Joel Berk and C. Scott Ananian
  - Lexical Analyzer Generator (JLex) - see LICENSE.txt

---

Apache Xerces Java  
Copyright 1999-2006 The Apache Software Foundation

This product includes software developed at  
The Apache Software Foundation (<http://www.apache.org/>).

Portions of Apache Xerces Java in xercesImpl.jar and xml-apis.jar were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc.,

software copyright (c) 1999.

---

Apache xml-commons xml-apis (redistribution of xml-apis.jar)

Apache XML Commons

Copyright 2001-2003,2006 The Apache Software Foundation.

This product includes software developed at

The Apache Software Foundation (<http://www.apache.org/>).

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., <http://www.ibm.com>.
- software copyright (c) 1999, Sun Microsystems., <http://www.sun.com>.
- software copyright (c) 2000 World Wide Web Consortium, <http://www.w3.org>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2016 Cisco Systems, Inc. All rights reserved.