Open Source Used In Webex App - Mac Webex App 46.1

Cisco Systems, Inc.
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.

Text Part Number: 78EE117C99-2872269937

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 submit this form.

In your requests please include the following reference number 78EE117C99-2872269937

Contents

1.1 websocketpp 0.8.0

1.1.1 Available under license :

Main Library:

Copyright (c) 2014, Peter Thorson. 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 WebSocket++ Project 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 PETER THORSON 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.

Bundled Libraries:

****** Base 64 Library (base64/base64.hpp) ******
base64.hpp is a repackaging of the base64.cpp and base64.h files into a
single header suitable for use as a header only library. This conversion was
done by Peter Thorson (webmaster@zaphoyd.com) in 2012. All modifications to
the code are redistributed under the same license as the original, which is
listed below.

base64.cpp and base64.h

Copyright (C) 2004-2008 Ren Nyffenegger

This source code is provided 'as-is',
without any express or implied
warranty. In no event will the author be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this source code must not be misrepresented; you must not
claim that you wrote the original source code. If you use this source code
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original source code.

3. This notice may not be removed or altered from any source distribution.

Ren Nyffenegger rene.nyffenegger@adp-gmbh.ch

****** SHA1 Library (sha1/sha1.hpp) ******
sha1.hpp is a repackaging of the sha1.cpp and sha1.h files from the shallsha1
library (http://code.google.com/p/smallsha1/)
 into a single header suitable for
use as a header only library. This conversion was done by Peter Thorson
(webmaster@zaphoyd.com) in 2013. All modifications to the code are redistributed
under the same license as the original, which is listed below.

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 Micael Hildenborg 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 Micael Hildenborg "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 Micael Hildenborg 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.

****** MD5 Library (common/md5.hpp) ******
md5.hpp is a reformulation of the md5.h and md5.c code from
http://www.opensource.apple.com/source/cups/cups-59/cups/md5.c to allow it to
function as a component of a header only library. This conversion
 was done by
Peter Thorson (webmaster@zaphoyd.com) in 2012 for the WebSocket++ project. The
changes are released under the same license as the original (listed below)

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not
be removed or altered from any source distribution.

L. Peter Deutsch
ghost@aladdin.com

****** UTF8 Validation logic (utf8_validation.hpp) ******
utf8_validation.hpp is adapted from code originally written by Bjoern Hoehrmann
<bjoern@hoehrmann.de>. See http://bjoern.hoehrmann.de/utf-8/decoder/dfa/ for
details.

The original license:

Copyright (c) 2008-2009 Bjoern Hoehrmann <bjoern@hoehrmann.de>

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.

1.2 tidy 5.7.24

1.2.1 Available under license :

HTML Tidy

HTML parser and pretty printer

Copyright (c) 1998-2016 World Wide Web Consortium
(Massachusetts Institute of Technology, European Research
Consortium for Informatics and Mathematics, Keio University).
All Rights Reserved.

Additional contributions (c) 2001-2016 University of Toronto, Terry Teague,
@geoffmcl, HTACG, and others.

Contributing Author(s):

Dave Raggett <dsr@w3.org>

The contributing author(s) would like to thank all those who
helped with testing, bug fixes and suggestions for improvements.
This wouldn't have been possible without your help.

COPYRIGHT NOTICE:

This software and documentation is provided "as is," and
the copyright holders and contributing author(s) 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.

The copyright holders and contributing author(s) will not be held
liable for any direct, indirect, special or consequential damages
arising out of any use of the software or documentation, even if
advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute
this source code, or portions hereof, documentation and executables,
for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must
not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any
source or altered source distribution.

The copyright holders and contributing author(s) specifically

permit, without fee, and encourage the use of this source code
as a component for supporting the Hypertext Markup Language in
commercial products. If you use this source code in a product,
acknowledgement is not
required but would be appreciated.

1.3 libusb 1.0.21

1.3.1 Available under license :

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, Fifth 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!

1.4 protobuf 3.7.1

1.5 jansson 2.12

1.5.1 Available under license :

Copyright (c) 2009-2018 Petri Lehtinen <petri@digip.org>

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.

1.6 opus 1.3.1

1.6.1 Available under license :

Copyright 2001-2011 Xiph.Org, Skype Limited, Octasic,
Jean-Marc Valin, Timothy B. Terriberry,
CSIRO, Gregory Maxwell, Mark Borgerding,
Erik de Castro Lopo

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 Internet Society, IETF or IETF Trust, nor the
names of specific 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.

Opus is subject to the royalty-free patent licenses which are
specified at:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Microsoft Corporation:
https://datatracker.ietf.org/ipr/1914/

Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/

1.7 libilbc 2.0.2

1.7.1 Available under license :

Copyright (c) 2011, The WebRTC project authors. 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 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 HOLDER 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.

1.8 json-cpp 0.7.0

1.8.1 Available under license :

The JsonCpp library's source code, including accompanying documentation, tests and demonstration applications, are licensed under the following conditions...

The author (Baptiste Lepilleur) explicitly disclaims copyright in all jurisdictions which recognize such a disclaimer. In such jurisdictions, this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of 2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur, and is released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this software may choose to accept it either as 1) Public Domain, 2) under the conditions of the MIT License (see below), or 3) under the terms of dual Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is described in clear, concise terms at:

http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

==========

Copyright (c) 2007-2010 Baptiste Lepilleur

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.

==========

(END LICENSE TEXT)

The MIT license is compatible with both the GPL and commercial software, affording one all of the rights of Public Domain with the minor nuisance of being required to keep the above copyright notice and license text in the source code. Note also that by accepting the Public Domain "license" you can re-license your copy using whatever license you like.

1.9 json-c 0.14

1.9.1 Available under license :

Copyright (c) 2009-2012 Eric Haszlakiewicz

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) 2004, 2005 Metaparadigm Pte Ltd

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.

1.10 openssl 1.1.1g

1.10.1 Available under license :

LICENSE ISSUES
==========

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License

/* ==
* Copyright (c) 1998-2019 The OpenSSL Project. 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. All advertising materials mentioning features or use
 of this
*   software must display the following acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*   endorse or promote products derived from this software without
*   prior written permission. For written permission, please contact
*   openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*   nor may "OpenSSL" appear in their names without prior written
*   permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``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 THE OpenSSL PROJECT OR
* ITS 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.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* 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 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. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* "This product includes cryptographic software written by
* Eric Young (eay@cryptsoft.com)"
* The word 'cryptographic' can be left out if the rouines from the library
* being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include
an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``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 AUTHOR 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.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this
code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

GNU GENERAL PUBLIC LICENSE
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.

GNU GENERAL PUBLIC LICENSE
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

Appendix: 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) 19yy <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) 19yy 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.

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis
of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it

under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent
notice in each changed file stating how and
when you changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable
form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided
that you do not advertise this Package as a
product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines
(or comparably compiled subroutines in other
languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT
LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

1.11 tiny-xml 2.5.3

1.11.1 Available under license :

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

/** @mainpage

<h1> TinyXML </h1>

TinyXML is a simple, small, C++ XML parser that can be easily
integrated into other programs.

<h2> What it does. </h2>

In brief, TinyXML parses an XML document, and builds from that a
Document Object Model (DOM) that can be read, modified, and saved.

XML stands for "eXtensible Markup Language." It allows you to create
your own document markups. Where HTML does a very good job of marking
documents for browsers, XML allows you to define any kind of document
markup, for example a document that describes a "to do" list for an
organizer application. XML is a very structured and convenient format.
All those random file formats created to store application data can
all be replaced with XML. One parser for everything.

The best place for the complete, correct, and quite frankly hard to
read spec is at
http://www.w3.org/TR/2004/REC-xml-20040204/.
 An intro to XML
(that I really like) can be found at
http://skew.org/xml/tutorial.

There are different ways to access and interact with XML data.
TinyXML uses a Document Object Model (DOM), meaning the XML data is parsed
into a C++ objects that can be browsed and manipulated, and then
written to disk or another output stream. You can also construct an XML document
from scratch with C++ objects and write this to disk or another output
stream.

TinyXML is designed to be easy and fast to learn. It is two headers
and four cpp files. Simply add these to your project and off you go.
There is an example file - xmltest.cpp - to get you started.

TinyXML is released under the ZLib license,
so you can use it in open source or commercial code. The details
of the license are at the top of every source file.

TinyXML attempts to be a flexible parser, but with truly correct and
compliant XML output. TinyXML should compile on
 any reasonably C++
compliant system. It does not rely on exceptions or RTTI. It can be
compiled with or without STL support. TinyXML fully supports
the UTF-8 encoding, and the first 64k character entities.

<h2> What it doesn't do. </h2>

TinyXML doesn't parse or use DTDs (Document Type Definitions) or XSLs
(eXtensible Stylesheet Language.) There are other parsers out there
(check out www.sourceforge.org, search for XML) that are much more fully
featured. But they are also much bigger, take longer to set up in
your project, have a higher learning curve, and often have a more
restrictive license. If you are working with browsers or have more
complete XML needs, TinyXML is not the parser for you.

The following DTD syntax will not parse at this time in TinyXML:

@verbatim
 <!DOCTYPE Archiv [
  <!ELEMENT Comment (#PCDATA)>
 ]>
@endverbatim

because TinyXML sees this as a !DOCTYPE node with an illegally
embedded !ELEMENT node. This may be addressed in the future.

<h2> Tutorials. </h2>

For
 the impatient, here is a tutorial to get you going. A great way to get started,
but it is worth your time to read this (very short) manual completely.

- @subpage tutorial0

<h2> Code Status. </h2>

TinyXML is mature, tested code. It is very stable. If you find
bugs, please file a bug report on the sourceforge web site
(www.sourceforge.net/projects/tinyxml). We'll get them straightened
out as soon as possible.

There are some areas of improvement; please check sourceforge if you are
interested in working on TinyXML.

<h2> Related Projects </h2>

TinyXML projects you may find useful! (Descriptions provided by the projects.)

 TinyXPath (http://tinyxpath.sourceforge.net). TinyXPath is a small footprint
XPath syntax decoder, written in C++.
 TinyXML++ (http://code.google.com/p/ticpp/). TinyXML++ is a completely new
interface to TinyXML that uses MANY of the C++ strengths. Templates,
exceptions, and much better error handling.

<h2>
Features </h2>

<h3> Using STL </h3>

TinyXML can be compiled to use or not use STL. When using STL, TinyXML
uses the std::string class, and fully supports std::istream, std::ostream,
operator<<, and operator>>. Many API methods have both 'const char*' and
'const std::string&' forms.

When STL support is compiled out, no STL files are included whatsoever. All
the string classes are implemented by TinyXML itself. API methods
all use the 'const char*' form for input.

Use the compile time #define:

TIXML_USE_STL

to compile one version or the other. This can be passed by the compiler,
or set as the first line of "tinyxml.h".

Note: If compiling the test code in Linux, setting the environment
variable TINYXML_USE_STL=YES/NO will control STL compilation. In the
Windows project file, STL and non STL targets are provided. In your project,
It's probably easiest to add the line "#define TIXML_USE_STL" as the first
line of tinyxml.h.

<h3> UTF-8 </h3>

TinyXML supports UTF-8 allowing
to manipulate XML files in any language. TinyXML
also supports "legacy mode" - the encoding used before UTF-8 support and
probably best described as "extended ascii".

Normally, TinyXML will try to detect the correct encoding and use it. However,
by setting the value of TIXML_DEFAULT_ENCODING in the header file, TinyXML

can be forced to always use one encoding.

TinyXML will assume Legacy Mode until one of the following occurs:

  If the non-standard but common "UTF-8 lead bytes" (0xef 0xbb 0xbf)
  begin the file or data stream, TinyXML will read it as UTF-8.
  If the declaration tag is read, and it has an encoding="UTF-8", then
  TinyXML will read it as UTF-8.
  If the declaration tag is read, and it has no encoding specified, then TinyXML will
  read it as UTF-8.
  If the declaration tag is read, and it has an encoding="something else", then TinyXML
  will read it as Legacy Mode. In legacy mode, TinyXML will work as it did before. It's

 not clear what that mode does exactly, but old content should keep working.
  Until one of the above criteria is met, TinyXML runs in Legacy Mode.

What happens if the encoding is incorrectly set or detected? TinyXML will try
to read and pass through text seen as improperly encoded. You may get some strange results or
mangled characters. You may want to force TinyXML to the correct mode.

You may force TinyXML to Legacy Mode by using LoadFile(TIXML_ENCODING_LEGACY) or
LoadFile(filename, TIXML_ENCODING_LEGACY). You may force it to use legacy mode all
the time by setting TIXML_DEFAULT_ENCODING = TIXML_ENCODING_LEGACY. Likewise, you may
force it to TIXML_ENCODING_UTF8 with the same technique.

For English users, using English XML, UTF-8 is the same as low-ASCII. You
don't need to be aware of UTF-8 or change your code in any way. You can think
of UTF-8 as a "superset" of ASCII.

UTF-8 is not a double byte format - but it is a standard encoding of Unicode!
TinyXML
 does not use or directly support wchar, TCHAR, or Microsoft's _UNICODE at this time.
It is common to see the term "Unicode" improperly refer to UTF-16, a wide byte encoding
of unicode. This is a source of confusion.

For "high-ascii" languages - everything not English, pretty much - TinyXML can
handle all languages, at the same time, as long as the XML is encoded
in UTF-8. That can be a little tricky, older programs and operating systems
tend to use the "default" or "traditional" code page. Many apps (and almost all
modern ones) can output UTF-8, but older or stubborn (or just broken) ones
still output text in the default code page.

For example, Japanese systems traditionally use SHIFT-JIS encoding.
Text encoded as SHIFT-JIS can not be read by TinyXML.
A good text editor can import SHIFT-JIS and then save as UTF-8.

The Skew.org link does a great

job covering the encoding issue.

The test file "utf8test.xml" is an XML containing English,
Spanish, Russian,
and Simplified Chinese. (Hopefully they are translated correctly). The file
"utf8test.gif" is a screen capture of the XML file, rendered in IE. Note that
if you don't have the correct fonts (Simplified Chinese or Russian) on your
system, you won't see output that matches the GIF file even if you can parse
it correctly. Also note that (at least on my Windows machine) console output
is in a Western code page, so that Print() or printf() cannot correctly display
the file. This is not a bug in TinyXML - just an OS issue. No data is lost or
destroyed by TinyXML. The console just doesn't render UTF-8.

<h3> Entities </h3>
TinyXML recognizes the pre-defined "character entities", meaning special
characters. Namely:

@verbatim
& &
< <
> >
" "
' '
@endverbatim

These are recognized when the XML document is read, and translated to there
UTF-8 equivalents. For instance, text with the XML of:

@verbatim
Far & Away
@endverbatim

will have the Value()
of "Far & Away" when queried from the TiXmlText object,
and will be written back to the XML stream/file as an ampersand. Older versions
of TinyXML "preserved" character entities, but the newer versions will translate
them into characters.

Additionally, any character can be specified by its Unicode code point:
The syntax " " or " " are both to the non-breaking space characher.

<h3> Printing </h3>
TinyXML can print output in several different ways that all have strengths and limitations.

- Print(FILE*). Output to a std-C stream, which includes all C files as well as stdout.
- "Pretty prints", but you don't have control over printing options.
- The output is streamed directly to the FILE object, so there is no memory overhead

in the TinyXML code.
- used by Print() and SaveFile()

- operator<<. Output to a c++ stream.
- Integrates with standart C++ iostreams.
- Outputs in "network printing" mode without line breaks. Good for network transmission and moving XML
between C++ objects, but hard for a human to read.

- TiXmlPrinter. Output to a std::string or memory buffer.
- API is less concise
- Future printing options will be put here.
- Printing may change slightly in future versions as it is refined and expanded.

<h3> Streams </h3>
With TIXML_USE_STL on TinyXML supports C++ streams (operator <<,>>) streams as well as C (FILE*) streams. There are some differences that you may need to be aware of.

C style output:
- based on FILE*
- the Print() and SaveFile() methods

Generates formatted output, with plenty of white space, intended to be as human-readable as possible. They are very fast, and tolerant of ill formed XML documents. For example, an XML document that contains 2 root elements and 2 declarations, will still print.

C style input:
- based on FILE*
- the Parse() and LoadFile() methods

A fast, tolerant read. Use whenever you don't need the C++ streams.

C++ style output:
- based on std::ostream
- operator<<

Generates condensed output, intended for network transmission rather than readability. Depending on your system's implementation of the ostream class, these may be somewhat slower. (Or may not.) Not tolerant of ill formed XML: a document should contain the correct one root element. Additional root level elements will not be streamed out.

C++ style input:
- based on std::istream
- operator>>

Reads XML from a stream, making it useful for network transmission. The tricky
part is knowing when the XML document is complete, since there will almost
certainly be other data in the stream. TinyXML will assume the XML data is
complete after it reads the root element. Put another way, documents that
are ill-constructed with more than one root element will not read correctly.
Also note that operator>> is somewhat slower than Parse, due to both
implementation of the STL and limitations of TinyXML.

<h3> White space </h3>
The world simply does not agree on whether white space should be kept,
or condensed.
For example, pretend the '_' is a space, and look at "Hello____world". HTML, and
at least some XML parsers, will interpret this as "Hello_world". They condense white
space. Some XML parsers do not, and will leave it as "Hello____world". (Remember
to keep pretending the _ is a space.) Others suggest that __Hello___world__ should become
Hello___world.

It's an issue that hasn't been resolved to my satisfaction. TinyXML supports the
first 2 approaches. Call TiXmlBase::SetCondenseWhiteSpace(bool) to set the desired behavior.
The default is to condense white space.

If you change the default, you should call TiXmlBase::SetCondenseWhiteSpace(bool)
before making any calls to Parse XML data, and I don't recommend changing it after
it has been set.

<h3> Handles </h3>

Where browsing an XML document in a robust way, it is important to check
for null returns from method calls. An error safe implementation can
generate a lot of code like:

@verbatim
TiXmlElement* root = document.FirstChildElement(
 "Document");
if (root)
{
 TiXmlElement* element = root->FirstChildElement("Element");
 if (element)
 {
  TiXmlElement* child = element->FirstChildElement("Child");
  if (child)
  {
   TiXmlElement* child2 = child->NextSiblingElement("Child");
   if (child2)
   {
    // Finally do something useful.
@endverbatim

Handles have been introduced to clean this up. Using the TiXmlHandle class,
the previous code reduces to:

@verbatim
TiXmlHandle docHandle(&document);
TiXmlElement* child2 = docHandle.FirstChild("Document").FirstChild("Element").Child("Child", 1
).ToElement();
if (child2)
{
 // do something useful
@endverbatim

Which is much easier to deal with. See TiXmlHandle for more information.

<h3> Row and Column tracking </h3>
Being able to track nodes and attributes back to their origin location
in source files can be very important for some applications. Additionally,
knowing where parsing errors occured in the original source can be very
time
 saving.

TinyXML can tracks the row and column origin of all nodes and attributes
in a text file. The TiXmlBase::Row() and TiXmlBase::Column() methods return
the origin of the node in the source text. The correct tabs can be
configured in TiXmlDocument::SetTabSize().

<h2> Using and Installing </h2>

To Compile and Run xmltest:

A Linux Makefile and a Windows Visual C++ .dsw file is provided.
Simply compile and run. It will write the file demotest.xml to your
disk and generate output on the screen. It also tests walking the
DOM by printing out the number of nodes found using different
techniques.

The Linux makefile is very generic and runs on many systems - it
is currently tested on mingw and
MacOSX. You do not need to run 'make depend'. The dependecies have been
hard coded.

<h3>Windows project file for VC6</h3>

tinyxml: tinyxml library, non-STL
tinyxmlSTL: tinyxml library, STL

tinyXmlTest: test app, non-STL
tinyXmlTestSTL: test app,
STL

<h3>Makefile</h3>
At the top of the makefile you can set:

PROFILE, DEBUG, and TINYXML_USE_STL. Details (such that they are) are in
the makefile.

In the tinyxml directory, type "make clean" then "make". The executable
file 'xmltest' will be created.

<h3>To Use in an Application:</h3>

Add tinyxml.cpp, tinyxml.h, tinyxmlerror.cpp, tinyxmlparser.cpp, tinystr.cpp, and tinystr.h to your
project or make file. That's it! It should compile on any reasonably
compliant C++ system. You do not need to enable exceptions or
RTTI for TinyXML.

<h2> How TinyXML works. </h2>

An example is probably the best way to go. Take:
@verbatim
<?xml version="1.0" standalone=no>
<!-- Our to do list data -->
<ToDo>
<Item priority="1"> Go to the <bold>Toy store!</bold></Item>
<Item priority="2"> Do bills</Item>
</ToDo>
@endverbatim

Its not much of a To Do list, but it will do. To read this file
(say "demo.xml") you would create a document, and parse it in:
@verbatim
TiXmlDocument
doc("demo.xml");
doc.LoadFile();
@endverbatim

And its ready to go. Now lets look at some lines and how they
relate to the DOM.

@verbatim

<?xml version="1.0" standalone=no>
@endverbatim

The first line is a declaration, and gets turned into the
TiXmlDeclaration class. It will be the first child of the
document node.

This is the only directive/special tag parsed by TinyXML.
Generally directive tags are stored in TiXmlUnknown so the
commands wont be lost when it is saved back to disk.

@verbatim
<!-- Our to do list data -->
@endverbatim

A comment. Will become a TiXmlComment object.

@verbatim
<ToDo>
@endverbatim

The "ToDo" tag defines a TiXmlElement object. This one does not have
any attributes, but does contain 2 other elements.

@verbatim
<Item priority="1">
@endverbatim

Creates another TiXmlElement which is a child of the "ToDo" element.
This element has 1 attribute, with the name "priority" and the value
"1".

@verbatim
Go to the
@endverbatim

A TiXmlText.
This is a leaf node and cannot contain other nodes.
It is a child of the "Item" TiXmlElement.

@verbatim
<bold>
@endverbatim

Another TiXmlElement, this one a child of the "Item" element.

Etc.

Looking at the entire object tree, you end up with:
@verbatim
TiXmlDocument "demo.xml"
 TiXmlDeclaration "version='1.0'" "standalone=no"
 TiXmlComment " Our to do list data"
 TiXmlElement "ToDo"
  TiXmlElement "Item" Attribtutes: priority = 1
   TiXmlText "Go to the "
   TiXmlElement "bold"
    TiXmlText "Toy store!"
  TiXmlElement "Item" Attributes: priority=2
   TiXmlText "Do bills"
@endverbatim

<h2> Documentation </h2>

The documentation is build with Doxygen, using the 'dox'
configuration file.

<h2> License </h2>

TinyXML is released under the zlib license:

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to
 anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

<h2> References </h2>

The World Wide Web Consortium is the definitive standard body for
XML, and their web pages contain huge amounts of information.

The definitive spec:
http://www.w3.org/TR/2004/REC-xml-20040204/

I also recommend "XML Pocket Reference" by Robert Eckstein and published by
OReilly...the book that got
the whole thing started.

<h2> Contributors, Contacts, and a Brief History </h2>

Thanks very much to everyone who sends suggestions, bugs, ideas, and
encouragement. It all helps, and makes this project fun. A special thanks
to the contributors on the web pages that keep it lively.

So many people have sent in bugs and ideas, that rather than list here
we try to give credit due in the "changes.txt" file.

TinyXML was originally written by Lee Thomason. (Often the "I" still
in the documentation.) Lee reviews changes and releases new versions,
with the help of Yves Berquin, Andrew Ellerton, and the tinyXml community.

We appreciate your suggestions, and would love to know if you
use TinyXML. Hopefully you will enjoy it and find it useful.
Please post questions, comments, file bugs, or contact us at:

www.sourceforge.net/projects/tinyxml

Lee Thomason, Yves Berquin, Andrew Ellerton
*/

Found in path(s):
* /tinyxml-2-6-2-1-tar-gz/tinyxml/readme.txt
No license file was found, but licenses were detected in source scan.

TinyXML is released under the ZLib license, so you can use it in open source or commercial code. The details of the license are at the top of every source file.
TinyXML is released under the zlib license:
Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:
1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Found in path(s):
* /tinyxml-2-6-2-1-tar-gz/tinyxml/docs/index.html

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

00008 Permission is granted to anyone to use this software for any
00012 1. The origin of this software must not be misrepresented; you must
00017 2. Altered source versions must be plainly marked as such, and
00020 3. This notice may not be removed or altered from any source

Found in path(s):
* /tinyxml-2-6-2-1-tar-gz/tinyxml/docs/tinystr_8h_source.html
No license file was found, but licenses were detected in source scan.

/*
www.sourceforge.net/projects/tinyxml
Original code (2.0 and earlier)copyright (c) 2000-2006 Lee Thomason (www.grinninglizard.com)

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.
*/

Found
in path(s):
* /tinyxml-2-6-2-1-tar-gz/tinyxml/tinyxmlerror.cpp
No license file was found, but licenses were detected in source scan.

/*
www.sourceforge.net/projects/tinyxml
Original code by Lee Thomason (www.grinninglizard.com)

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.
*/

Found in path(s):
* /tinyxml-2-6-2-1-tar-gz/tinyxml/tinyxml.cpp
*
/tinyxml-2-6-2-1-tar-gz/tinyxml/tinyxml.h
No license file was found, but licenses were detected in source scan.

/*
www.sourceforge.net/projects/tinyxml
Original code by Lee Thomason (www.grinninglizard.com)

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

*/

Found in path(s):
*
/tinyxml-2-6-2-1-tar-gz/tinyxml/tinyxmlparser.cpp
No license file was found, but licenses were detected in source scan.

/*
www.sourceforge.net/projects/tinyxml

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.
*/

Found in path(s):
* /tinyxml-2-6-2-1-tar-gz/tinyxml/tinystr.cpp
* /tinyxml-2-6-2-1-tar-gz/tinyxml/tinystr.h
No license file was found, but licenses were detected in source scan.

00009 Permission is granted to anyone to use this software for any
00013 1. The origin of this software must not be misrepresented; you must
00018 2. Altered source versions must be plainly marked as such, and
00021 3. This notice may not be removed or altered from any source

Found in path(s):
* /tinyxml-2-6-2-1-tar-gz/tinyxml/docs/tinyxml_8h_source.html

1.12 openssl 1.1.1q

1.12.1 Available under license :

LICENSE ISSUES
==========

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License

/* ==
* Copyright (c) 1998-2019 The OpenSSL Project. 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. All advertising materials mentioning features or use
 of this
*   software must display the following acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*   endorse or promote products derived from this software without
*   prior written permission. For written permission, please contact
*   openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*   nor may "OpenSSL" appear in their names without prior written
*   permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project

*   for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``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 THE OpenSSL PROJECT OR
* ITS 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.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
 *
* 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 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. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include
 an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``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 AUTHOR 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.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this
 code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

GNU GENERAL PUBLIC LICENSE
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.

GNU GENERAL PUBLIC LICENSE
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

Appendix: 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) 19yy <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) 19yy 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.

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis
of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item

itself, though there may be fees involved in handling the item.
It also means that recipients of the item may redistribute it
under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the
Standard Version of this Package without restriction, provided that you
duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications
derived from the Public Domain or from the Copyright Holder. A Package
modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided
that you insert a prominent
notice in each changed file stating how and
when you changed that file, and provided that you do at least ONE of the
following:

a) place your modifications in the Public Domain or otherwise make them
Freely Available, such as by posting said modifications to Usenet or
an equivalent medium, or placing the modifications on a major archive
site such as uunet.uu.net, or by allowing the Copyright Holder to include
your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict
with standard executables, which must also be provided, and provide
a separate manual page for each non-standard executable that clearly
documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or
executable
form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files,
together with instructions (in the manual page or equivalent) on where
to get the Standard Version.

b) accompany the distribution with the machine-readable source of
the Package with your modifications.

c) give non-standard executables non-standard names, and clearly
document the differences in manual pages (or equivalent), together
with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided
that you do not advertise this Package as a
product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines
(or comparably compiled subroutines in other
languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT
LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

1.13 safestring 4.1.3

1.13.1 Available under license :

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

```
/*
 Tokenizing SafeStrings and converting to numbers
 Examples of how to use the nextToken() and toLong() and toDouble() to parse a CSV line

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeString_nextToken/SafeString_nextToken.ino
No license file was found, but licenses were detected in source scan.

```
/*
 Appending to SafeStrings using print()/println()
 Examples of how to use the standard print()/println() to append and format different data types to SafeStrings

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringPrint/SafeStringPrint.ino
No license file was found, but licenses were detected in source scan.

```
/*
 SafeString removeFrom(), removeBefore(), remove(), removeLast(), keepLast()
 Examples of SafeString remove

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.
```

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringRemove/SafeStringRemove.ino
No license file was found, but licenses were detected in source scan.

/*
SafeString constructors and assignments
Examples of how to create SafeStrings and how to assign SafeStrings from other data types
also see the SafeStringFromCharArray, SafeStringFromCharPtr and SafeStringFromCharPtrWithSize examples

by Matthew Ford
Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
Modified from String Examples by Tom Igoe
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeString_ConstructorAndDebugging/SafeString_ConstructorAndDebugging.ino
No license file was found, but licenses were detected in source scan.

/*
SafeString.cpp V2.0.0 static memory SafeString library modified by
Matthew Ford
Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
All rights reservered subject to the License below

extensively modified from

WString.cpp - SafeString library for Wiring & Arduino
...mostly rewritten by Paul Stoffregen...
Copyright (c) 2009-10 Hernando Barragan. All rights reserved.
Copyright 2011, Paul Stoffregen, paul@pjrc.com

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 St, Fifth Floor, Boston, MA 02110-1301 USA
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/src/SafeString.cpp
No license file was found, but licenses were detected in source scan.

/*
SafeString replace()
Examples of SafeString replace for chars and strings

by Matthew Ford
Copyright(c)2020 Forward Computing and Control Pty. Ltd.
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringReplace/SafeStringReplace.ino
No license file was found, but licenses were detected in source scan.

/*
Test Errors for Assignment to SafeStrings using the = operator

by Matthew Ford
Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
Modified from String Examples by Tom Igoe
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-
4.1.24/examples/SafeString_Tests/SafeStringEquAssignErrTests/SafeStringEquAssignErrTests.ino
No license file was found, but licenses were detected in source scan.

/*
SafeString readUntil, non-blocking until delimiter found
Example of how to use the non-blocking readUntil() method to parse a CSV line

by Matthew Ford

Copyright(c)2020 Forward Computing and Control Pty. Ltd.
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-
4.1.24/examples/SafeString_Tests/SafeString_readUntil/SafeString_readUntil.ino
No license file was found, but licenses were detected in source scan.

/*
SafeStringReader_GPS.ino

This example reads GPS data from a SafeStringStream continuously using a SafeStringReader

by Matthew Ford
Copyright(c)2020 Forward Computing and Control Pty. Ltd.
This example code is in the public domain.

download and install the SafeString library from Arduino library manager
or from www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeStringReader_GPS/SafeStringReader_GPS.ino
No license file was found, but licenses were detected in source scan.

/*
Prefixing to SafeStrings using the -= operator and prefix()
Examples of how to prefix different data types to SafeStrings

by Matthew Ford
Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
Modified from String Examples by Tom Igoe
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-
4.1.24/examples/SafeString_Tests/SafeStringPrefixOperator/SafeStringPrefixOperator.ino
No license file was found, but licenses were detected in source scan.

/* SafeStringInClasses_1.ino
Example of using SafeString for Class char[]s

by Matthew Ford
Copyright(c)2020 Forward Computing and Control Pty. Ltd.
This example code is in the public domain.

download and install the SafeString library from Arduino library manager
or from www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeStringsInClasses_1/SafeStringsInClasses_1.ino
No license file was found, but licenses were detected in source scan.

/*
SSring indexOf() and lastIndexOf()

by Matthew Ford
Copyright(c)2020 Forward Computing and Control Pty. Ltd.
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringIndexOf/SafeStringIndexOf.ino
No license file was found, but licenses were detected in source scan.

/*
SafeString from char* with specified size constructor
Examples of how to create SafeStrings from an existing pointer to a char[]
also see the SafeString_ConstructorAndDebugging, SafeStringFromCharArray and SafeStringFromCharPtr examples

by Matthew Ford
Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
Modified from String Examples by Tom Igoe
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringFromCharPtrWithSize/SafeStringFromCharPtrWithSize.ino
No license file was found, but licenses were detected in source scan.

/*
SafeString Case changes
Examples of how to change the case of a SafeString

by Matthew Ford
Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
Modified from String Examples by Tom Igoe
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringCaseChanges/SafeStringCaseChanges.ino
No license file was found, but licenses were detected in source scan.

/*
SafeString.h static memory SafeString library modified by
Matthew Ford
Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
All rights reserved subject to the License below

modified from
WString.h - String library for Wiring & Arduino
...mostly rewritten by Paul Stoffregen...
Copyright (c) 2009-10 Hernando Barragan. All right reserved.
Copyright 2011, Paul Stoffregen, paul@pjrc.com

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 St, Fifth Floor, Boston, MA 02110-1301 USA
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/src/SafeString.h
No license file was found, but licenses were detected in source scan.

/*
OBD_Processor.ino for SafeString V4.0.0+

Example of using SafeString to process Car OnBoardData

by Matthew Ford
Copyright(c)2020 Forward Computing and Control Pty. Ltd.
This example code is in the public domain.

download and install the SafeString library V4.0.0+ from Arduino library manager
or from www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/OBD_Processor/OBD_Processor.ino
No license file was found, but licenses were detected in source scan.

/*
SafeString from char* constructor
Examples of how to create SafeStrings from an existing pointer to a char[]
also see the SafeString_ConstructorAndDebugging, SafeStringFromCharArray and
SafeStringFromCharPtrWithSize examples

by Matthew Ford
Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
Modified from String Examples by Tom Igoe
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-
4.1.24/examples/SafeString_Tests/SafeStringFromCharPtr/SafeStringFromCharPtr.ino
No license file was found, but licenses were detected in source scan.

/*
SafeString substring()
Examples of SafeString substring

by Matthew Ford
Copyright(c)2020 Forward Computing and Control Pty. Ltd.
This example code is in the public domain.

www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringSubstring/SafeStringSubstring.ino

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

```
/* SafeStringWithArraysOfCstrings.ino
   Example of using SafeString for working with char[][xx]

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 download and install the SafeString library from Arduino library manager
 or from www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeStringWithArraysOfCstrings/SafeStringWithArraysOfCstrings.ino
No license file was found, but licenses were detected in source scan.

```
/*
 SafeStringReader_flushInput.ino

 This example flushes any initial input and also starts flushing if "flush" is found in the text stream

 by Matthew Ford
 Copyright(c)2021 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 download and install the SafeString library from Arduino library manager
 or from www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeStringReader_flushInput/SafeStringReader_flushInput.ino
No license file was found, but licenses were detected in source scan.

```
/*
 SafeString charAt() and setCharAt()
 Examples of how to get and set characters of a SafeString

 by Matthew Ford
 Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 Modified from String Examples by Tom Igoe
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):

* /safestring-4-1-24-zip/SafeString-
4.1.24/examples/SafeString_Tests/SafeStringCharacters/SafeStringCharacters.ino
No license file was found, but licenses were detected in source scan.

```
/*
 Comparing SafeStrings
 Examples of how to compare SafeStrings using the comparison operators

 by Matthew Ford
 Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 Modified from String Examples by Tom Igoe
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):
* /safestring-4-1-24-zip/SafeString-
4.1.24/examples/SafeString_Tests/SafeStringComparisonOperators/SafeStringComparisonOperators.ino
No license file was found, but licenses were detected in source scan.

```
/*
 SafeString to Number conversion
 Examples of SafeString to Number conversions and comparing these to the results from String methods

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringToNum/SafeStringToNum.ino
No license file was found, but licenses were detected in source scan.

```
/*
 Fixed 9 Formatting of double/float/long/ing

 by Matthew Ford
 Copyright(c)2021 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):
* /safestring-4-1-24-zip/SafeString-

4.1.24/examples/SafeString_Tests/SafeString_fixedWidthFormat/SafeString_fixedWidthFormat.ino
No license file was found, but licenses were detected in source scan.

/*
 SafeSting length(), trim()

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringLengthTrim/SafeStringLengthTrim.ino
No license file was found, but licenses were detected in source scan.

/*
 SafeStringReader_Assign.ino

 This example = to SafeStringReader

 by Matthew Ford
 Copyright(c)2021 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 download and install the SafeString library from Arduino library manager
 or from www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeStringReader_Assign/SafeStringReader_Assign.ino
No license file was found, but licenses were detected in source scan.

/*
 SafeString from char[] constructor
 Examples of how to create SafeStrings from an existing char[]
 also see the SafeString_ConstructorAndDebugging, SafeStringFromCharPtr and SafeStringFromCharPtrWithSize examples

 by Matthew Ford
 Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 Modified from String Examples by Tom Igoe
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html

*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-
4.1.24/examples/SafeString_Tests/SafeStringFromCharArray/SafeStringFromCharArray.ino
No license file was found, but licenses were detected in source scan.

/*
 Tokenizing SafeStrings and converting to numbers
 Examples of how to use the stoken and toLong() and toDouble() to parse a CSV line

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeString_stoken/SafeString_stoken.ino
No license file was found, but licenses were detected in source scan.

/*
 sfStream.ino

 SafeStringStream Unit test
 Example of using SafeStringStream

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 https://www.forward.com.au/pfod/ArduinoProgramming/Serial_IO/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeStringStream_Tests/sfStream/sfStream.ino
No license file was found, but licenses were detected in source scan.

/*
 SafeString startsWith(), startsWithIgnoreCase(), endsWithCharFrom() and endsWith()

 by Matthew Ford
 Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringStartsWithEndsWith/SafeStringStartsWithEndsWith.ino
No license file was found, but licenses were detected in source scan.

```
/*
 Blink

 Turns an led on for one second, then off for one second, repeatedly.

 Most Arduinos have an on-board led you can control. On the UNO, MEGA and ZERO
 it is attached to digital pin 13, on MKR1000 on pin 6. led is set to
 the correct led pin independent of which board is used.
 If you want to know what pin the on-board led is connected to on your Arduino
 model, check the Technical Specs of your board at:
 https://www.arduino.cc/en/Main/Products

 modified 8 May 2014
 by Scott Fitzgerald
 modified 2 Sep 2016
 by Arturo Guadalupi
 modified 8 Sep 2016
 by Colby Newman

 This example code is in the public domain.

 http://www.arduino.cc/en/Tutorial/Blink
*/
```

Found in path(s):
* /safestring-4-1-24-zip/SafeString-4.1.24/examples/loopTimer/LoopTimer_BlinkDelay/LoopTimer_BlinkDelay.ino
No license file was found, but licenses were detected in source scan.

```
/*
 Appending to SafeStrings using the += operator and concat()
 Examples of how to append different data types to SafeStrings
 Also has examples of using hasError() method

 by Matthew Ford
 Mods Copyright(c)2020 Forward Computing and Control Pty. Ltd.
 Modified from String Examples by Tom Igoe
 This example code is in the public domain.

 www.forward.com.au/pfod/ArduinoProgramming/SafeString/index.html
*/
```

Found in path(s):

* /safestring-4-1-24-zip/SafeString-4.1.24/examples/SafeString_Tests/SafeStringAssignmentAndConcatOperator/SafeStringAssignmentAndConcatOperator.ino

1.14 cpprest 2.9.0

1.14.1 Available under license :

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

```
$$ -*- mode: c++; -*-
$$ This is a Pump source file.  Please use Pump to convert it to
$$ gmock-generated-function-mockers.h.
$$
$var n = 10  $$ The maximum arity we support.
// Copyright 2007, 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.
//
// Author: wan@google.com (Zhanyong Wan)

// Google Mock - a framework for writing C++ mock classes.
```

```
//
// This file contains template meta-programming utility classes needed
// for implementing Google
 Mock.

#ifndef GMOCK_INCLUDE_GMOCK_INTERNAL_GMOCK_GENERATED_INTERNAL_UTILS_H_
#define GMOCK_INCLUDE_GMOCK_INTERNAL_GMOCK_GENERATED_INTERNAL_UTILS_H_

#include "gmock/internal/gmock-port.h"

namespace testing {

template <typename T>
class Matcher;

namespace internal {

// An IgnoredValue object can be implicitly constructed from ANY value.
// This is used in implementing the IgnoreResult(a) action.
class IgnoredValue {
public:
 // This constructor template allows any value to be implicitly
 // converted to IgnoredValue.  The object has no data member and
 // doesn't try to remember anything about the argument.  We
 // deliberately omit the 'explicit' keyword in order to allow the
 // conversion to be implicit.
 template <typename T>
 IgnoredValue(const T& /* ignored */) {}  // NOLINT(runtime/explicit)
};

// MatcherTuple<T>::type is a tuple type where each field is a Matcher
// for the corresponding field in tuple type T.
template <typename Tuple>
struct MatcherTuple;


$range
 i 0..n
$for i [[
$range j 1..i
$var typename_As = [[$for j, [[typename A$j]]]]
$var As = [[$for j, [[A$j]]]]
$var matcher_As = [[$for j, [[Matcher<A$j>]]]]
template <$typename_As>
struct MatcherTuple< ::std::tr1::tuple<$As> > {
 typedef ::std::tr1::tuple<$matcher_As > type;
};
```

```
]]
// Template struct Function<F>, where F must be a function type, contains
// the following typedefs:
//
//   Result:              the function's return type.
//   ArgumentN:           the type of the N-th argument, where N starts with 1.
//   ArgumentTuple:       the tuple type consisting of all parameters of F.
//   ArgumentMatcherTuple: the tuple type consisting of Matchers for all
//                        parameters of F.
//   MakeResultVoid:      the function type obtained by substituting void
//                        for the return type of F.
//   MakeResultIgnoredValue:
//                        the function type obtained by substituting Something
//                        for the return
 type of F.
template <typename F>
struct Function;

template <typename R>
struct Function<R()> {
 typedef R Result;
 typedef ::std::tr1::tuple<> ArgumentTuple;
 typedef typename MatcherTuple<ArgumentTuple>::type ArgumentMatcherTuple;
 typedef void MakeResultVoid();
 typedef IgnoredValue MakeResultIgnoredValue();
};


$range i 1..n
$for i [[
$range j 1..i
$var typename_As = [[$for j [[, typename A$j]]]]
$var As = [[$for j, [[A$j]]]]
$var matcher_As = [[$for j, [[Matcher<A$j>]]]]
$range k 1..i-1
$var prev_As = [[$for k, [[A$k]]]]
template <typename R$typename_As>
struct Function<R($As)>
  : Function<R($prev_As)> {
 typedef A$i Argument$i;
 typedef ::std::tr1::tuple<$As> ArgumentTuple;
 typedef typename MatcherTuple<ArgumentTuple>::type ArgumentMatcherTuple;
 typedef void MakeResultVoid($As);
 typedef IgnoredValue MakeResultIgnoredValue($As);
};
```

]]
} // namespace internal

} // namespace testing

#endif // GMOCK_INCLUDE_GMOCK_INTERNAL_GMOCK_GENERATED_INTERNAL_UTILS_H_

Found
 in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/internal/gmock-generated-internal-utils.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/internal/gmock-generated-internal-utils.h.pump
No license file was found, but licenses were detected in source scan.

$$ -*- mode: c++; -*-
$$ This is a Pump source file. Please use Pump to convert it to
$$ gmock-generated-function-mockers.h.
$$
$var n = 10 $$ The maximum arity we support.
// Copyright 2007, 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.

```
//
// Author: wan@google.com (Zhanyong Wan)

// Google Mock - a framework for writing C++ mock classes.
//
// This file implements function mockers of various arities.

#ifndef GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_FUNCTION_MOCKERS_H_
#define
 GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_FUNCTION_MOCKERS_H_

#include "gmock/gmock-spec-builders.h"
#include "gmock/internal/gmock-internal-utils.h"

namespace testing {
namespace internal {

template <typename F>
class FunctionMockerBase;

// Note: class FunctionMocker really belongs to the ::testing
// namespace.  However if we define it in ::testing, MSVC will
// complain when classes in ::testing::internal declare it as a
// friend class template.  To workaround this compiler bug, we define
// FunctionMocker in ::testing::internal and import it into ::testing.
template <typename F>
class FunctionMocker;


$range i 0..n
$for i [[
$range j 1..i
$var typename_As = [[$for j [[, typename A$j]]]]
$var As = [[$for j, [[A$j]]]]
$var as = [[$for j, [[a$j]]]]
$var Aas = [[$for j, [[A$j a$j]]]]
$var ms = [[$for j, [[m$j]]]]
$var matchers = [[$for j, [[const Matcher<A$j>& m$j]]]]
template <typename R$typename_As>
class FunctionMocker<R($As)> : public
  internal::FunctionMockerBase<R($As)>
 {
public:
 typedef R F($As);
 typedef typename internal::Function<F>::ArgumentTuple ArgumentTuple;

 MockSpec<F>& With($matchers) {
```

```
$if i >= 1 [[
  this->current_spec().SetMatchers(::std::tr1::make_tuple($ms));

]]
  return this->current_spec();
 }

 R Invoke($Aas) {
  // Even though gcc and MSVC don't enforce it, 'this->' is required
  // by the C++ standard [14.6.4] here, as the base class type is
  // dependent on the template argument (and thus shouldn't be
  // looked into when resolving InvokeWith).
  return this->InvokeWith(ArgumentTuple($as));
 }
};



]]
}  // namespace internal

// The style guide prohibits "using" statements in a namespace scope
// inside a header file.  However, the FunctionMocker class template
// is meant to be defined in the ::testing namespace.  The following
// line is just a trick for working around a bug in MSVC 8.0, which
// cannot handle it if we define FunctionMocker in ::testing.
using internal::FunctionMocker;

//
 GMOCK_RESULT_(tn, F) expands to the result type of function type F.
// We define this as a variadic macro in case F contains unprotected
// commas (the same reason that we use variadic macros in other places
// in this file).
// INTERNAL IMPLEMENTATION - DON'T USE IN USER CODE!!!
#define GMOCK_RESULT_(tn, ...) \
  tn ::testing::internal::Function<__VA_ARGS__>::Result

// The type of argument N of the given function type.
// INTERNAL IMPLEMENTATION - DON'T USE IN USER CODE!!!
#define GMOCK_ARG_(tn, N, ...) \
  tn ::testing::internal::Function<__VA_ARGS__>::Argument##N

// The matcher type for argument N of the given function type.
// INTERNAL IMPLEMENTATION - DON'T USE IN USER CODE!!!
#define GMOCK_MATCHER_(tn, N, ...) \
  const ::testing::Matcher<GMOCK_ARG_(tn, N, __VA_ARGS__)>&

// The variable for mocking the given method.
// INTERNAL IMPLEMENTATION - DON'T USE IN USER CODE!!!
```

```
#define GMOCK_MOCKER_(arity, constness, Method) \
  GTEST_CONCAT_TOKEN_(gmock##constness##arity##_##Method##_,
 __LINE__)


$for i [[
$range j 1..i
$var arg_as = [[$for j, \
    [[GMOCK_ARG_(tn, $j, __VA_ARGS__) gmock_a$j]]]]
$var as = [[$for j, [[gmock_a$j]]]]
$var matcher_as = [[$for j, \
                [[GMOCK_MATCHER_(tn, $j, __VA_ARGS__) gmock_a$j]]]]
// INTERNAL IMPLEMENTATION - DON'T USE IN USER CODE!!!
#define GMOCK_METHOD$i[[]]_(tn, constness, ct, Method, ...) \
 GMOCK_RESULT_(tn, __VA_ARGS__) ct Method( \
    $arg_as) constness { \
  GTEST_COMPILE_ASSERT_((::std::tr1::tuple_size<                        \
      tn ::testing::internal::Function<__VA_ARGS__>::ArgumentTuple>::value == $i), \
      this_method_does_not_take_$i[[]]_argument[[$if i != 1 [[s]]]]); \
  GMOCK_MOCKER_($i, constness, Method).SetOwnerAndName(this, #Method); \
  return GMOCK_MOCKER_($i, constness, Method).Invoke($as); \
 } \
 ::testing::MockSpec<__VA_ARGS__>& \
    gmock_##Method($matcher_as) constness { \
  GMOCK_MOCKER_($i, constness, Method).RegisterOwner(this);
 \
  return GMOCK_MOCKER_($i, constness, Method).With($as); \
 } \
 mutable ::testing::FunctionMocker<__VA_ARGS__> GMOCK_MOCKER_($i, constness, Method)


]]
$for i [[
#define MOCK_METHOD$i(m, ...) GMOCK_METHOD$i[[]]_(, , , m, __VA_ARGS__)

]]


$for i [[
#define MOCK_CONST_METHOD$i(m, ...) GMOCK_METHOD$i[[]]_(, const, , m, __VA_ARGS__)

]]


$for i [[
#define MOCK_METHOD$i[[]]_T(m, ...) GMOCK_METHOD$i[[]]_(typename, , , m, __VA_ARGS__)

]]
```

```
$for i [[
#define MOCK_CONST_METHOD$i[[]]_T(m, ...) \
  GMOCK_METHOD$i[[]]_(typename, const, , m, __VA_ARGS__)

]]


$for i [[
#define MOCK_METHOD$i[[]]_WITH_CALLTYPE(ct, m, ...) \
  GMOCK_METHOD$i[[]]_(, , ct, m, __VA_ARGS__)

]]


$for i [[
#define MOCK_CONST_METHOD$i[[]]_WITH_CALLTYPE(ct, m, ...) \
  GMOCK_METHOD$i[[]]_(, const, ct, m, __VA_ARGS__)

]]


$for i [[
#define MOCK_METHOD$i[[]]_T_WITH_CALLTYPE(ct, m, ...) \
  GMOCK_METHOD$i[[]]_(typename, , ct, m, __VA_ARGS__)

]]


$for i [[
#define MOCK_CONST_METHOD$i[[]]_T_WITH_CALLTYPE(ct,
 m, ...) \
  GMOCK_METHOD$i[[]]_(typename, const, ct, m, __VA_ARGS__)

]]

// A MockFunction<F> class has one mock method whose type is F.  It is
// useful when you just want your test code to emit some messages and
// have Google Mock verify the right messages are sent (and perhaps at
// the right times).  For example, if you are exercising code:
//
//   Foo(1);
//   Foo(2);
//   Foo(3);
//
// and want to verify that Foo(1) and Foo(3) both invoke
// mock.Bar("a"), but Foo(2) doesn't invoke anything, you can write:
```

```
//
// TEST(FooTest, InvokesBarCorrectly) {
//   MyMock mock;
//   MockFunction<void(string check_point_name)> check;
//   {
//     InSequence s;
//
//     EXPECT_CALL(mock, Bar("a"));
//     EXPECT_CALL(check, Call("1"));
//     EXPECT_CALL(check, Call("2"));
//     EXPECT_CALL(mock, Bar("a"));
//   }
//   Foo(1);
//   check.Call("1");
//   Foo(2);
//   check.Call("2");
//   Foo(3);
// }
//
// The expectation spec says that the first Bar("a") must happen
//
 before check point "1", the second Bar("a") must happen after check
// point "2", and nothing should happen between the two check
// points. The explicit check points make it easy to tell which
// Bar("a") is called by which call to Foo().
template <typename F>
class MockFunction;


$for i [[
$range j 0..i-1
template <typename R$for j [[, typename A$j]]>
class MockFunction<R($for j, [[A$j]])> {
public:
 MockFunction() {}

 MOCK_METHOD$i[[]]_T(Call, R($for j, [[A$j]]));

private:
 GTEST_DISALLOW_COPY_AND_ASSIGN_(MockFunction);
};


]]
}  // namespace testing

#endif  // GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_FUNCTION_MOCKERS_H_
```

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-generated-function-mockers.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-generated-function-mockers.h.pump
No license file was found, but licenses were detected in source scan.

```
// All Rights Reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * 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
//     * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
```

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-param-util.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-param-util.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-param-util-generated.h
*
/vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-param-util-generated.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest-typed-test.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-type-util.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest-typed-test.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-type-util.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-tuple.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-tuple.h
No license file was found, but licenses were detected in source scan.

```
// All rights reserved.
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
//     * 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
//     * Neither the name of Google Inc. nor the names of its
// this software without specific prior written permission.
```

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-generated-actions.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest-message.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest-death-test.h
*

/vendors-gtest-1-8-0-234-windows-zip/include/gmock/internal/gmock-port.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-port.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/internal/gmock-internal-utils.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-generated-actions.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-cardinalities.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-generated-function-mockers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-generated-function-mockers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-filepath.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-actions.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-matchers.h
*
/vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-cardinalities.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-more-matchers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-generated-nice-strict.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/internal/gmock-generated-internal-utils.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-filepath.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-more-actions.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-port.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest-test-part.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest_pred_impl.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest-printers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest-param-test.h
*
/vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-generated-matchers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest_pred_impl.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-matchers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-internal.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/internal/gmock-port.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest-spi.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-internal.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-spec-builders.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest-printers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest-death-test.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock.h
*
/vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-linked_ptr.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-string.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-string.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-generated-matchers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/internal/gmock-internal-utils.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest-message.h

* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest-test-part.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest-param-test.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-generated-nice-strict.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/internal/gmock-generated-internal-utils.h
*
 /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest_prod.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest-spi.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest_prod.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-linked_ptr.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-more-matchers.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-death-test-internal.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-actions.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-spec-builders.h
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-death-test-internal.h
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-more-actions.h
No license file was found, but licenses were detected in source scan.

$$ -*- mode: c++; -*-
$$ This is a Pump source file. Please use Pump to convert it to
$$ gmock-generated-actions.h.
$$
$var n = 10 $$ The maximum arity we support.
$$}} This meta comment fixes auto-indentation in editors.
// Copyright 2007, 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.
//
// Author: wan@google.com (Zhanyong Wan)

// Google Mock - a framework for writing C++ mock classes.
//
// This file implements some commonly used
 variadic actions.

#ifndef GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_ACTIONS_H_
#define GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_ACTIONS_H_

#include "gmock/gmock-actions.h"
#include "gmock/internal/gmock-port.h"

namespace testing {
namespace internal {

// InvokeHelper<F> knows how to unpack an N-tuple and invoke an N-ary
// function or method with the unpacked values, where F is a function
// type that takes N arguments.
template <typename Result, typename ArgumentTuple>
class InvokeHelper;


$range i 0..n
$for i [[
$range j 1..i
$var types = [[$for j [[, typename A$j]]]]
$var as = [[$for j, [[A$j]]]]
$var args = [[$if i==0 [[]] $else [[ args]]]]
$var import = [[$if i==0 [[]] $else [[
  using ::std::tr1::get;

]]]]
$var gets = [[$for j, [[get<$(j - 1)>(args)]]]]
template <typename R$types>
class InvokeHelper<R, ::std::tr1::tuple<$as> > {
public:
 template <typename Function>
 static R Invoke(Function function, const ::std::tr1::tuple<$as>&$args) {
$import    return function($gets);
```

```
 }

  template <class Class, typename MethodPtr>
 static R InvokeMethod(Class* obj_ptr,
                       MethodPtr method_ptr,
                       const ::std::tr1::tuple<$as>&$args) {
$import    return (obj_ptr->*method_ptr)($gets);
 }
};



]]
// CallableHelper has static methods for invoking "callables",
// i.e. function pointers and functors.  It uses overloading to
// provide a uniform interface for invoking different kinds of
// callables.  In particular, you can use:
//
//   CallableHelper<R>::Call(callable, a1, a2, ..., an)
//
// to invoke an n-ary callable, where R is its return type.  If an
// argument, say a2, needs to be passed by reference, you should write
// ByRef(a2) instead of a2 in the above expression.
template <typename R>
class CallableHelper {
public:
 // Calls a nullary callable.
 template <typename Function>
 static R Call(Function function) { return function(); }

 // Calls a unary callable.

 // We deliberately pass a1 by value instead of const reference
 here
 // in case it is a C-string literal.  If we had declared the
 // parameter as 'const A1& a1' and write Call(function, "Hi"), the
 // compiler would've thought A1 is 'char[3]', which causes trouble
 // when you need to copy a value of type A1.  By declaring the
 // parameter as 'A1 a1', the compiler will correctly infer that A1
 // is 'const char*' when it sees Call(function, "Hi").
 //
 // Since this function is defined inline, the compiler can get rid
 // of the copying of the arguments.  Therefore the performance won't
 // be hurt.
 template <typename Function, typename A1>
 static R Call(Function function, A1 a1) { return function(a1); }

$range i 2..n
$for i
```

```
[[
$var arity = [[$if i==2 [[binary]] $elif i==3 [[ternary]] $else [[$i-ary]]]]

 // Calls a $arity callable.

$range j 1..i
$var typename_As = [[$for j, [[typename A$j]]]]
$var Aas = [[$for j, [[A$j a$j]]]]
$var as = [[$for j, [[a$j]]]]
$var typename_Ts = [[$for j, [[typename T$j]]]]
$var Ts = [[$for
 j, [[T$j]]]]
 template <typename Function, $typename_As>
 static R Call(Function function, $Aas) {
   return function($as);
 }

]]
};  // class CallableHelper

// An INTERNAL macro for extracting the type of a tuple field.  It's
// subject to change without notice - DO NOT USE IN USER CODE!
#define GMOCK_FIELD_(Tuple, N) \
   typename ::std::tr1::tuple_element<N, Tuple>::type

$range i 1..n

// SelectArgs<Result, ArgumentTuple, k1, k2, ..., k_n>::type is the
// type of an n-ary function whose i-th (1-based) argument type is the
// k{i}-th (0-based) field of ArgumentTuple, which must be a tuple
// type, and whose return type is Result.  For example,
//   SelectArgs<int, ::std::tr1::tuple<bool, char, double, long>, 0, 3>::type
// is int(bool, long).
//
// SelectArgs<Result, ArgumentTuple, k1, k2, ..., k_n>::Select(args)
// returns the selected fields (k1, k2, ..., k_n) of args as a tuple.
// For example,
//   SelectArgs<int, ::std::tr1::tuple<bool, char, double>, 2, 0>::Select(
//
       ::std::tr1::make_tuple(true, 'a', 2.5))
// returns ::std::tr1::tuple (2.5, true).
//
// The numbers in list k1, k2, ..., k_n must be >= 0, where n can be
// in the range [0, $n].  Duplicates are allowed and they don't have
// to be in an ascending or descending order.

template <typename Result, typename ArgumentTuple, $for i, [[int k$i]]>
class SelectArgs {
```

```
public:
 typedef Result type($for i, [[GMOCK_FIELD_(ArgumentTuple, k$i)]]);
 typedef typename Function<type>::ArgumentTuple SelectedArgs;
 static SelectedArgs Select(const ArgumentTuple& args) {
   using ::std::tr1::get;
   return SelectedArgs($for i, [[get<k$i>(args)]]);
 }
};


$for i [[
$range j 1..n
$range j1 1..i-1
template <typename Result, typename ArgumentTuple$for j1[[, int k$j1]]>
class SelectArgs<Result, ArgumentTuple,
                 $for j, [[$if j <= i-1 [[k$j]] $else [[-1]]]]> {
public:
 typedef Result type($for j1, [[GMOCK_FIELD_(ArgumentTuple, k$j1)]]);
 typedef typename Function<type>::ArgumentTuple
 SelectedArgs;
 static SelectedArgs Select(const ArgumentTuple& [[]]
$if i == 1 [[/* args */]] $else [[args]]) {
   using ::std::tr1::get;
   return SelectedArgs($for j1, [[get<k$j1>(args)]]);
 }
};


]]
#undef GMOCK_FIELD_

$var ks = [[$for i, [[k$i]]]]

// Implements the WithArgs action.
template <typename InnerAction, $for i, [[int k$i = -1]]>
class WithArgsAction {
public:
 explicit WithArgsAction(const InnerAction& action) : action_(action) {}

 template <typename F>
 operator Action<F>() const { return MakeAction(new Impl<F>(action_)); }

private:
 template <typename F>
 class Impl : public ActionInterface<F> {
  public:
   typedef typename Function<F>::Result Result;
   typedef typename Function<F>::ArgumentTuple ArgumentTuple;
```

```
  explicit Impl(const InnerAction& action) : action_(action) {}

  virtual Result Perform(const ArgumentTuple& args) {
    return action_.Perform(SelectArgs<Result, ArgumentTuple, $ks>::Select(args));
  }

 private:
  typedef
 typename SelectArgs<Result, ArgumentTuple,
      $ks>::type InnerFunctionType;

  Action<InnerFunctionType> action_;
 };

 const InnerAction action_;

 GTEST_DISALLOW_ASSIGN_(WithArgsAction);
};

// A macro from the ACTION* family (defined later in this file)
// defines an action that can be used in a mock function.  Typically,
// these actions only care about a subset of the arguments of the mock
// function.  For example, if such an action only uses the second
// argument, it can be used in any mock function that takes >= 2
// arguments where the type of the second argument is compatible.
//
// Therefore, the action implementation must be prepared to take more
// arguments than it needs.  The ExcessiveArg type is used to
// represent those excessive arguments.  In order to keep the compiler
// error messages tractable, we define it in the testing namespace
// instead of testing::internal.  However, this is an INTERNAL TYPE
// and subject to change without notice, so a user MUST
 NOT USE THIS
// TYPE DIRECTLY.
struct ExcessiveArg {};

// A helper class needed for implementing the ACTION* macros.
template <typename Result, class Impl>
class ActionHelper {
public:
$range i 0..n
$for i

[[
$var template = [[$if i==0 [[]] $else [[
$range j 0..i-1
 template <$for j, [[typename A$j]]>
```

```
]]]]
$range j 0..i-1
$var As = [[$for j, [[A$j]]]]
$var as = [[$for j, [[get<$j>(args)]]]]
$range k 1..n-i
$var eas = [[$for k, [[ExcessiveArg()]]]]
$var arg_list = [[$if (i==0) | (i==n) [[$as$eas]] $else [[$as, $eas]]]]
$template
 static Result Perform(Impl* impl, const ::std::tr1::tuple<$As>& args) {
   using ::std::tr1::get;
   return impl->template gmock_PerformImpl<$As>(args, $arg_list);
 }

]]
};

}  // namespace internal

// Various overloads for Invoke().

// WithArgs<N1, N2, ..., Nk>(an_action) creates an action that passes
// the selected arguments of the mock function to an_action and
// performs it.  It serves as an adaptor between actions with
// different argument
 lists.  C++ doesn't support default arguments for
// function templates, so we have to overload it.

$range i 1..n
$for i [[
$range j 1..i
template <$for j [[int k$j, ]]typename InnerAction>
inline internal::WithArgsAction<InnerAction$for j [[, k$j]]>
WithArgs(const InnerAction& action) {
 return internal::WithArgsAction<InnerAction$for j [[, k$j]]>(action);
}



]]
// Creates an action that does actions a1, a2, ..., sequentially in
// each invocation.
$range i 2..n
$for i [[
$range j 2..i
$var types = [[$for j, [[typename Action$j]]]]
$var Aas = [[$for j [[, Action$j a$j]]]]

template <typename Action1, $types>
$range k 1..i-1
```

```
inline $for k [[internal::DoBothAction<Action$k, ]]Action$i$for k  [[>]]

DoAll(Action1 a1$Aas) {
$if i==2 [[

 return internal::DoBothAction<Action1, Action2>(a1, a2);
]] $else [[
$range j2 2..i

 return DoAll(a1, DoAll($for j2, [[a$j2]]));
]]

}

]]

}  // namespace testing

// The ACTION* family of macros can be used in a namespace scope to
// define custom
 actions easily.  The syntax:
//
//   ACTION(name) { statements; }
//
// will define an action with the given name that executes the
// statements.  The value returned by the statements will be used as
// the return value of the action.  Inside the statements, you can
// refer to the K-th (0-based) argument of the mock function by
// 'argK', and refer to its type by 'argK_type'.  For example:
//
//   ACTION(IncrementArg1) {
//     arg1_type temp = arg1;
//     return ++(*temp);
//   }
//
// allows you to write
//
//   ...WillOnce(IncrementArg1());
//
// You can also refer to the entire argument tuple and its type by
// 'args' and 'args_type', and refer to the mock function type and its
// return type by 'function_type' and 'return_type'.
//
// Note that you don't need to specify the types of the mock function
// arguments.  However rest assured that your code is still type-safe:
// you'll get a compiler error if *arg1 doesn't support the ++
// operator, or if the type of ++(*arg1) isn't
```

```
 compatible with the
// mock function's return type, for example.
//
// Sometimes you'll want to parameterize the action.   For that you can use
// another macro:
//
//   ACTION_P(name, param_name) { statements; }
//
// For example:
//
//   ACTION_P(Add, n) { return arg0 + n; }
//
// will allow you to write:
//
//   ...WillOnce(Add(5));
//
// Note that you don't need to provide the type of the parameter
// either.  If you need to reference the type of a parameter named
// 'foo', you can write 'foo_type'.  For example, in the body of
// ACTION_P(Add, n) above, you can write 'n_type' to refer to the type
// of 'n'.
//
// We also provide ACTION_P2, ACTION_P3, ..., up to ACTION_P$n to support
// multi-parameter actions.
//
// For the purpose of typing, you can view
//
//   ACTION_Pk(Foo, p1, ..., pk) { ... }
//
// as shorthand for
//
//   template <typename p1_type, ..., typename pk_type>
//   FooActionPk<p1_type, ..., pk_type> Foo(p1_type p1, ..., pk_type pk) { ... }
//
// In particular,
 you can provide the template type arguments
// explicitly when invoking Foo(), as in Foo<long, bool>(5, false);
// although usually you can rely on the compiler to infer the types
// for you automatically.  You can assign the result of expression
// Foo(p1, ..., pk) to a variable of type FooActionPk<p1_type, ...,
// pk_type>.  This can be useful when composing actions.
//
// You can also overload actions with different numbers of parameters:
//
//   ACTION_P(Plus, a) { ... }
//   ACTION_P2(Plus, a, b) { ... }
//
// While it's tempting to always use the ACTION* macros when defining
```

```
// a new action, you should also consider implementing ActionInterface
// or using MakePolymorphicAction() instead, especially if you need to
// use the action a lot.  While these approaches require more work,
// they give you more control on the types of the mock function
// arguments and the action parameters, which in general leads to
// better compiler error messages that pay off in the long run.  They
//
 also allow overloading actions based on parameter types (as opposed
// to just based on the number of parameters).
//
// CAVEAT:
//
// ACTION*() can only be used in a namespace scope.  The reason is
// that C++ doesn't yet allow function-local types to be used to
// instantiate templates.  The up-coming C++0x standard will fix this.
// Once that's done, we'll consider supporting using ACTION*() inside
// a function.
//
// MORE INFORMATION:
//
// To learn more about using these macros, please search for 'ACTION'
// on http://code.google.com/p/googlemock/wiki/CookBook.

$range i 0..n
$range k 0..n-1

// An internal macro needed for implementing ACTION*().
#define GMOCK_ACTION_ARG_TYPES_AND_NAMES_UNUSED_\
  const args_type& args GTEST_ATTRIBUTE_UNUSED_
$for k [[, \
  arg$k[[]]_type arg$k GTEST_ATTRIBUTE_UNUSED_]]


// Sometimes you want to give an action explicit template parameters
// that cannot be inferred from its value parameters.  ACTION() and
// ACTION_P*() don't support that.
  ACTION_TEMPLATE() remedies that
// and can be viewed as an extension to ACTION() and ACTION_P*().
//
// The syntax:
//
//   ACTION_TEMPLATE(ActionName,
//                   HAS_m_TEMPLATE_PARAMS(kind1, name1, ..., kind_m, name_m),
//                   AND_n_VALUE_PARAMS(p1, ..., p_n)) { statements; }
//
// defines an action template that takes m explicit template
// parameters and n value parameters.  name_i is the name of the i-th
// template parameter, and kind_i specifies whether it's a typename,
```

```
// an integral constant, or a template.  p_i is the name of the i-th
// value parameter.
//
// Example:
//
//   // DuplicateArg<k, T>(output) converts the k-th argument of the mock
//   // function to type T and copies it to *output.
//   ACTION_TEMPLATE(DuplicateArg,
//                   HAS_2_TEMPLATE_PARAMS(int, k, typename, T),
//                   AND_1_VALUE_PARAMS(output)) {
//     *output = T(std::tr1::get<k>(args));
//   }
//   ...
//     int n;
//     EXPECT_CALL(mock, Foo(_,
 _))
//         .WillOnce(DuplicateArg<1, unsigned char>(&n));
//
// To create an instance of an action template, write:
//
//   ActionName<t1, ..., t_m>(v1, ..., v_n)
//
// where the ts are the template arguments and the vs are the value
// arguments.  The value argument types are inferred by the compiler.
// If you want to explicitly specify the value argument types, you can
// provide additional template arguments:
//
//   ActionName<t1, ..., t_m, u1, ..., u_k>(v1, ..., v_n)
//
// where u_i is the desired type of v_i.
//
// ACTION_TEMPLATE and ACTION/ACTION_P* can be overloaded on the
// number of value parameters, but not on the number of template
// parameters.  Without the restriction, the meaning of the following
// is unclear:
//
//   OverloadedAction<int, bool>(x);
//
// Are we using a single-template-parameter action where 'bool' refers
// to the type of x, or are we using a two-template-parameter action
// where the compiler is asked to infer the type of x?
//
// Implementation
 notes:
//
// GMOCK_INTERNAL_*_HAS_m_TEMPLATE_PARAMS and
// GMOCK_INTERNAL_*_AND_n_VALUE_PARAMS are internal macros for
// implementing ACTION_TEMPLATE.  The main trick we use is to create
```

```
// new macro invocations when expanding a macro.  For example, we have
//
//   #define ACTION_TEMPLATE(name, template_params, value_params)
//       ... GMOCK_INTERNAL_DECL_##template_params ...
//
// which causes ACTION_TEMPLATE(..., HAS_1_TEMPLATE_PARAMS(typename, T), ...)
// to expand to
//
//       ... GMOCK_INTERNAL_DECL_HAS_1_TEMPLATE_PARAMS(typename, T) ...
//
// Since GMOCK_INTERNAL_DECL_HAS_1_TEMPLATE_PARAMS is a macro, the
// preprocessor will continue to expand it to
//
//       ... typename T ...
//
// This technique conforms to the C++ standard and is portable.  It
// allows us to implement action templates using O(N) code, where N is
// the maximum number of template/value parameters supported.  Without
// using it, we'd have to devote O(N^2) amount of code to implement all
// combinations
 of m and n.

// Declares the template parameters.

$range j 1..n
$for j [[
$range m 0..j-1
#define GMOCK_INTERNAL_DECL_HAS_$j[[]]
_TEMPLATE_PARAMS($for m, [[kind$m, name$m]]) $for m, [[kind$m name$m]]



]]

// Lists the template parameters.

$for j [[
$range m 0..j-1
#define GMOCK_INTERNAL_LIST_HAS_$j[[]]
_TEMPLATE_PARAMS($for m, [[kind$m, name$m]]) $for m, [[name$m]]



]]

// Declares the types of value parameters.

$for i [[
$range j 0..i-1
#define GMOCK_INTERNAL_DECL_TYPE_AND_$i[[]]
```

```
_VALUE_PARAMS($for j, [[p$j]]) $for j [[, typename p$j##_type]]


]]

// Initializes the value parameters.

$for i [[
$range j 0..i-1
#define GMOCK_INTERNAL_INIT_AND_$i[[]]_VALUE_PARAMS($for j, [[p$j]])\
  ($for j, [[p$j##_type gmock_p$j]])$if i>0 [[ : ]]$for j, [[p$j(gmock_p$j)]]


]]

// Declares the fields for storing the value parameters.

$for i [[
$range j 0..i-1
#define GMOCK_INTERNAL_DEFN_AND_$i[[]]
_VALUE_PARAMS($for j, [[p$j]]) $for j [[p$j##_type p$j; ]]


]]

// Lists the value parameters.

$for
 i [[
$range j 0..i-1
#define GMOCK_INTERNAL_LIST_AND_$i[[]]
_VALUE_PARAMS($for j, [[p$j]]) $for j, [[p$j]]


]]

// Lists the value parameter types.

$for i [[
$range j 0..i-1
#define GMOCK_INTERNAL_LIST_TYPE_AND_$i[[]]
_VALUE_PARAMS($for j, [[p$j]]) $for j [[, p$j##_type]]


]]

// Declares the value parameters.
```

```
$for i [[
$range j 0..i-1
#define GMOCK_INTERNAL_DECL_AND_$i[[]]_VALUE_PARAMS($for j, [[p$j]]) [[]]
$for j, [[p$j##_type p$j]]


]]

// The suffix of the class template implementing the action template.
$for i [[


$range j 0..i-1
#define GMOCK_INTERNAL_COUNT_AND_$i[[]]_VALUE_PARAMS($for j, [[p$j]]) [[]]
$if i==1 [[P]] $elif i>=2 [[P$i]]
]]


// The name of the class template implementing the action template.
#define GMOCK_ACTION_CLASS_(name, value_params)\
  GTEST_CONCAT_TOKEN_(name##Action, GMOCK_INTERNAL_COUNT_##value_params)

$range k 0..n-1

#define ACTION_TEMPLATE(name, template_params, value_params)\
 template <GMOCK_INTERNAL_DECL_##template_params\

       GMOCK_INTERNAL_DECL_TYPE_##value_params>\
 class GMOCK_ACTION_CLASS_(name, value_params) {\
  public:\
   GMOCK_ACTION_CLASS_(name, value_params)\
     GMOCK_INTERNAL_INIT_##value_params {}\
   template <typename F>\
   class gmock_Impl : public ::testing::ActionInterface<F> {\
    public:\
     typedef F function_type;\
     typedef typename ::testing::internal::Function<F>::Result return_type;\
     typedef typename ::testing::internal::Function<F>::ArgumentTuple\
         args_type;\
     explicit gmock_Impl GMOCK_INTERNAL_INIT_##value_params {}\
     virtual return_type Perform(const args_type& args) {\
       return ::testing::internal::ActionHelper<return_type, gmock_Impl>::\
           Perform(this, args);\
     }\
     template <$for k, [[typename arg$k[[]]_type]]>\
     return_type gmock_PerformImpl(const args_type& args[[]]
$for k [[, arg$k[[]]_type arg$k]]) const;\
     GMOCK_INTERNAL_DEFN_##value_params\
```

```
  private:\
   GTEST_DISALLOW_ASSIGN_(gmock_Impl);\
  };\
  template <typename F> operator ::testing::Action<F>() const {\
   return ::testing::Action<F>(\
       new gmock_Impl<F>(GMOCK_INTERNAL_LIST_##value_params));\
  }\
  GMOCK_INTERNAL_DEFN_##value_params\
 private:\
  GTEST_DISALLOW_ASSIGN_(GMOCK_ACTION_CLASS_(name, value_params));\
};\
template <GMOCK_INTERNAL_DECL_##template_params\
          GMOCK_INTERNAL_DECL_TYPE_##value_params>\
inline GMOCK_ACTION_CLASS_(name, value_params)<\
    GMOCK_INTERNAL_LIST_##template_params\
    GMOCK_INTERNAL_LIST_TYPE_##value_params> name(\
        GMOCK_INTERNAL_DECL_##value_params) {\
  return GMOCK_ACTION_CLASS_(name, value_params)<\
      GMOCK_INTERNAL_LIST_##template_params\
      GMOCK_INTERNAL_LIST_TYPE_##value_params>(\
          GMOCK_INTERNAL_LIST_##value_params);\
}\
template <GMOCK_INTERNAL_DECL_##template_params\
          GMOCK_INTERNAL_DECL_TYPE_##value_params>\
template <typename F>\
template <typename arg0_type,
typename arg1_type, typename arg2_type, \
    typename arg3_type, typename arg4_type, typename arg5_type, \
    typename arg6_type, typename arg7_type, typename arg8_type, \
    typename arg9_type>\
typename ::testing::internal::Function<F>::Result\
    GMOCK_ACTION_CLASS_(name, value_params)<\
        GMOCK_INTERNAL_LIST_##template_params\
        GMOCK_INTERNAL_LIST_TYPE_##value_params>::gmock_Impl<F>::\
            gmock_PerformImpl(\
        GMOCK_ACTION_ARG_TYPES_AND_NAMES_UNUSED_) const

$for i

[[
$var template = [[$if i==0 [[]] $else [[
$range j 0..i-1

 template <$for j, [[typename p$j##_type]]>\
]]]]
$var class_name = [[name##Action[[$if i==0 [[]] $elif i==1 [[P]]
                                    $else [[P$i]]]]]]
$range j 0..i-1
```

```
$var ctor_param_list = [[$for j, [[p$j##_type gmock_p$j]]]]
$var param_types_and_names = [[$for j, [[p$j##_type p$j]]]]
$var inits = [[$if i==0 [[]] $else [[ : $for j, [[p$j(gmock_p$j)]]]]]]
$var param_field_decls
 = [[$for j
[[

    p$j##_type p$j;\
]]]]
$var param_field_decls2 = [[$for j
[[

  p$j##_type p$j;\
]]]]
$var params = [[$for j, [[p$j]]]]
$var param_types = [[$if i==0 [[]] $else [[<$for j, [[p$j##_type]]>]]]]
$var typename_arg_types = [[$for k, [[typename arg$k[[]]_type]]]]
$var arg_types_and_names = [[$for k, [[arg$k[[]]_type arg$k]]]]
$var macro_name = [[$if i==0 [[ACTION]] $elif i==1 [[ACTION_P]]
                                $else [[ACTION_P$i]]]]

#define $macro_name(name$for j [[, p$j]])\$template
 class $class_name {\
  public:\
   $class_name($ctor_param_list)$inits {}\
   template <typename F>\
   class gmock_Impl : public ::testing::ActionInterface<F> {\
    public:\
     typedef F function_type;\
     typedef typename ::testing::internal::Function<F>::Result return_type;\
     typedef typename ::testing::internal::Function<F>::ArgumentTuple\
         args_type;\
     [[$if i==1 [[explicit ]]]]gmock_Impl($ctor_param_list)$inits {}\
     virtual return_type
 Perform(const args_type& args) {\
       return ::testing::internal::ActionHelper<return_type, gmock_Impl>::\
           Perform(this, args);\
     }\
     template <$typename_arg_types>\
     return_type gmock_PerformImpl(const args_type& args, [[]]
$arg_types_and_names) const;\$param_field_decls
    private:\
     GTEST_DISALLOW_ASSIGN_(gmock_Impl);\
   };\
   template <typename F> operator ::testing::Action<F>() const {\
     return ::testing::Action<F>(new gmock_Impl<F>($params));\
   }\$param_field_decls2
  private:\
```

```
  GTEST_DISALLOW_ASSIGN_($class_name);\
 };\$template
 inline $class_name$param_types name($param_types_and_names) {\
  return $class_name$param_types($params);\
 }\$template
 template <typename F>\
 template <$typename_arg_types>\
 typename ::testing::internal::Function<F>::Result\
    $class_name$param_types::gmock_Impl<F>::gmock_PerformImpl(\
       GMOCK_ACTION_ARG_TYPES_AND_NAMES_UNUSED_) const
]]
$$ }  // This meta comment fixes
 auto-indentation in Emacs.  It won't
$$    // show up in the generated code.


namespace testing {

// The ACTION*() macros trigger warning C4100 (unreferenced formal
// parameter) in MSVC with -W4.  Unfortunately they cannot be fixed in
// the macro definition, as the warnings are generated when the macro
// is expanded and macro expansion cannot contain #pragma.  Therefore
// we suppress them here.
#ifdef _MSC_VER
# pragma warning(push)
# pragma warning(disable:4100)
#endif

// Various overloads for InvokeArgument<N>().
//
// The InvokeArgument<N>(a1, a2, ..., a_k) action invokes the N-th
// (0-based) argument, which must be a k-ary callable, of the mock
// function, with arguments a1, a2, ..., a_k.
//
// Notes:
//
//   1. The arguments are passed by value by default.  If you need to
//   pass an argument by reference, wrap it inside ByRef().  For
//   example,
//
//     InvokeArgument<1>(5, string("Hello"), ByRef(foo))
//
//   passes 5 and string("Hello") by value, and passes foo
 by
//   reference.
//
//   2. If the callable takes an argument by reference but ByRef() is
//   not used, it will receive the reference to a copy of the value,
```

```
//   instead of the original value.  For example, when the 0-th
//   argument of the mock function takes a const string&, the action
//
//     InvokeArgument<0>(string("Hello"))
//
//   makes a copy of the temporary string("Hello") object and passes a
//   reference of the copy, instead of the original temporary object,
//   to the callable.  This makes it easy for a user to define an
//   InvokeArgument action from temporary values and have it performed
//   later.

$range i 0..n
$for i [[
$range j 0..i-1

ACTION_TEMPLATE(InvokeArgument,
                HAS_1_TEMPLATE_PARAMS(int, k),
                AND_$i[[]]_VALUE_PARAMS($for j, [[p$j]])) {
 return internal::CallableHelper<return_type>::Call(
     ::std::tr1::get<k>(args)$for j [[, p$j]]);
}

]]

// Various overloads for ReturnNew<T>().
//
// The ReturnNew<T>(a1, a2, ...,
 a_k) action returns a pointer to a new
// instance of type T, constructed on the heap with constructor arguments
// a1, a2, ..., and a_k. The caller assumes ownership of the returned value.
$range i 0..n
$for i [[
$range j 0..i-1
$var ps = [[$for j, [[p$j]]]]

ACTION_TEMPLATE(ReturnNew,
                HAS_1_TEMPLATE_PARAMS(typename, T),
                AND_$i[[]]_VALUE_PARAMS($ps)) {
 return new T($ps);
}

]]

#ifdef _MSC_VER
# pragma warning(pop)
#endif

}  // namespace testing
```

#endif // GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_ACTIONS_H_

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-generated-actions.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-generated-actions.h.pump
No license file was found, but licenses were detected in source scan.

$$ -*- mode: c++; -*-
$var n = 50 $$ Maximum length of type lists we want to support.
// 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.
//
// Author: wan@google.com (Zhanyong Wan)

// Type utilities needed for implementing typed and type-parameterized
// tests. This file is generated by a SCRIPT. DO NOT EDIT BY HAND!
//
// Currently we support at most $n types in a list, and at most $n
// type-parameterized tests in one

```
 type-parameterized test case.
// Please contact googletestframework@googlegroups.com if you need
// more.

#ifndef GTEST_INCLUDE_GTEST_INTERNAL_GTEST_TYPE_UTIL_H_
#define GTEST_INCLUDE_GTEST_INTERNAL_GTEST_TYPE_UTIL_H_

#include "gtest/internal/gtest-port.h"

// #ifdef __GNUC__ is too general here.  It is possible to use gcc without using
// libstdc++ (which is where cxxabi.h comes from).
# if GTEST_HAS_CXXABI_H_
#  include <cxxabi.h>
# elif defined(__HP_aCC)
#  include <acxx_demangle.h>
# endif  // GTEST_HASH_CXXABI_H_

namespace testing {
namespace internal {

// GetTypeName<T>() returns a human-readable name of type T.
// NB: This function is also used in Google Mock, so don't move it inside of
// the typed-test-only section below.
template <typename T>
std::string GetTypeName() {
# if GTEST_HAS_RTTI

 const char* const name = typeid(T).name();
#  if GTEST_HAS_CXXABI_H_ || defined(__HP_aCC)
 int status = 0;
 // gcc's implementation of typeid(T).name() mangles the type name,
 //
 so we have to demangle it.
#   if GTEST_HAS_CXXABI_H_
 using abi::__cxa_demangle;
#   endif  // GTEST_HAS_CXXABI_H_
 char* const readable_name = __cxa_demangle(name, 0, 0, &status);
 const std::string name_str(status == 0 ? readable_name : name);
 free(readable_name);
 return name_str;
#  else
 return name;
#  endif  // GTEST_HAS_CXXABI_H_ || __HP_aCC

# else

 return "<type>";
```

```
# endif  // GTEST_HAS_RTTI
}

#if GTEST_HAS_TYPED_TEST || GTEST_HAS_TYPED_TEST_P

// AssertyTypeEq<T1, T2>::type is defined iff T1 and T2 are the same
// type.  This can be used as a compile-time assertion to ensure that
// two types are equal.

template <typename T1, typename T2>
struct AssertTypeEq;

template <typename T>
struct AssertTypeEq<T, T> {
 typedef bool type;
};

// A unique type used as the default value for the arguments of class
// template Types.  This allows us to simulate variadic templates
// (e.g. Types<int>, Type<int, double>, and etc), which C++ doesn't
// support directly.
struct
 None {};

// The following family of struct and struct templates are used to
// represent type lists.  In particular, TypesN<T1, T2, ..., TN>
// represents a type list with N types (T1, T2, ..., and TN) in it.
// Except for Types0, every struct in the family has two member types:
// Head for the first type in the list, and Tail for the rest of the
// list.

// The empty type list.
struct Types0 {};

// Type lists of length 1, 2, 3, and so on.

template <typename T1>
struct Types1 {
 typedef T1 Head;
 typedef Types0 Tail;
};

$range i 2..n

$for i [[
$range j 1..i
$range k 2..i
template <$for j, [[typename T$j]]>
```

```
struct Types$i {
 typedef T1 Head;
 typedef Types$(i-1)<$for k, [[T$k]]> Tail;
};


]]

}  // namespace internal

// We don't want to require the users to write TypesN<...> directly,
// as that would require them to count the length.  Types<...> is much
// easier to write, but generates horrible messages when there is a
// compiler error, as gcc insists on printing out each
 template
// argument, even if it has the default value (this means Types<int>
// will appear as Types<int, None, None, ..., None> in the compiler
// errors).
//
// Our solution is to combine the best part of the two approaches: a
// user would write Types<T1, ..., TN>, and Google Test will translate
// that to TypesN<T1, ..., TN> internally to make error messages
// readable.  The translation is done by the 'type' member of the
// Types template.

$range i 1..n
template <$for i, [[typename T$i = internal::None]]>
struct Types {
 typedef internal::Types$n<$for i, [[T$i]]> type;
};

template <>
struct Types<$for i, [[internal::None]]> {
 typedef internal::Types0 type;
};

$range i 1..n-1
$for i [[
$range j 1..i
$range k i+1..n
template <$for j, [[typename T$j]]>
struct Types<$for j, [[T$j]]$for k[[, internal::None]]> {
 typedef internal::Types$i<$for j, [[T$j]]> type;
};

]]

namespace internal {
```

```
# define GTEST_TEMPLATE_ template <typename T> class

// The template "selector" struct
 TemplateSel<Tmpl> is used to
// represent Tmpl, which must be a class template with one type
// parameter, as a type.  TemplateSel<Tmpl>::Bind<T>::type is defined
// as the type Tmpl<T>.  This allows us to actually instantiate the
// template "selected" by TemplateSel<Tmpl>.
//
// This trick is necessary for simulating typedef for class templates,
// which C++ doesn't support directly.
template <GTEST_TEMPLATE_ Tmpl>
struct TemplateSel {
 template <typename T>
 struct Bind {
   typedef Tmpl<T> type;
 };
};

# define GTEST_BIND_(TmplSel, T) \
 TmplSel::template Bind<T>::type

// A unique struct template used as the default value for the
// arguments of class template Templates.  This allows us to simulate
// variadic templates (e.g. Templates<int>, Templates<int, double>,
// and etc), which C++ doesn't support directly.
template <typename T>
struct NoneT {};

// The following family of struct and struct templates are used to
// represent template lists.  In particular, TemplatesN<T1,
 T2, ...,
// TN> represents a list of N templates (T1, T2, ..., and TN).  Except
// for Templates0, every struct in the family has two member types:
// Head for the selector of the first template in the list, and Tail
// for the rest of the list.

// The empty template list.
struct Templates0 {};

// Template lists of length 1, 2, 3, and so on.

template <GTEST_TEMPLATE_ T1>
struct Templates1 {
 typedef TemplateSel<T1> Head;
 typedef Templates0 Tail;
};
```

```
$range i 2..n

$for i [[
$range j 1..i
$range k 2..i
template <$for j, [[GTEST_TEMPLATE_ T$j]]>
struct Templates$i {
 typedef TemplateSel<T1> Head;
 typedef Templates$(i-1)<$for k, [[T$k]]> Tail;
};


]]

// We don't want to require the users to write TemplatesN<...> directly,
// as that would require them to count the length.  Templates<...> is much
// easier to write, but generates horrible messages when there is a
// compiler error, as gcc insists on printing out each template
// argument, even if it has the default value (this
 means Templates<list>
// will appear as Templates<list, NoneT, NoneT, ..., NoneT> in the compiler
// errors).
//
// Our solution is to combine the best part of the two approaches: a
// user would write Templates<T1, ..., TN>, and Google Test will translate
// that to TemplatesN<T1, ..., TN> internally to make error messages
// readable.  The translation is done by the 'type' member of the
// Templates template.

$range i 1..n
template <$for i, [[GTEST_TEMPLATE_ T$i = NoneT]]>
struct Templates {
 typedef Templates$n<$for i, [[T$i]]> type;
};

template <>
struct Templates<$for i, [[NoneT]]> {
 typedef Templates0 type;
};

$range i 1..n-1
$for i [[
$range j 1..i
$range k i+1..n
template <$for j, [[GTEST_TEMPLATE_ T$j]]>
struct Templates<$for j, [[T$j]]$for k[[, NoneT]]> {
 typedef Templates$i<$for j, [[T$j]]> type;
```

```
};

]]

// The TypeList template makes it possible to use either a single type
// or a Types<...> list in TYPED_TEST_CASE() and
// INSTANTIATE_TYPED_TEST_CASE_P().

template
 <typename T>
struct TypeList {
 typedef Types1<T> type;
};


$range i 1..n
template <$for i, [[typename T$i]]>
struct TypeList<Types<$for i, [[T$i]]> > {
 typedef typename Types<$for i, [[T$i]]>::type type;
};

#endif  // GTEST_HAS_TYPED_TEST || GTEST_HAS_TYPED_TEST_P

}  // namespace internal
}  // namespace testing

#endif  // GTEST_INCLUDE_GTEST_INTERNAL_GTEST_TYPE_UTIL_H_
```

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-type-util.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-type-util.h.pump
No license file was found, but licenses were detected in source scan.

```
$$ -*- mode: c++; -*-
$var n = 10  $$ Maximum number of tuple fields we want to support.
$$ This meta comment fixes auto-indentation in Emacs. }}
// Copyright 2009 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.
//
// Author: wan@google.com (Zhanyong Wan)

// Implements a subset of TR1 tuple needed by Google Test and Google Mock.

#ifndef GTEST_INCLUDE_GTEST_INTERNAL_GTEST_TUPLE_H_
#define GTEST_INCLUDE_GTEST_INTERNAL_GTEST_TUPLE_H_

#include
 <utility>  // For ::std::pair.

// The compiler used in Symbian has a bug that prevents us from declaring the
// tuple template as a friend (it complains that tuple is redefined).  This
// hack bypasses the bug by declaring the members that should otherwise be
// private as public.
// Sun Studio versions < 12 also have the above bug.
#if defined(__SYMBIAN32__) || (defined(__SUNPRO_CC) && __SUNPRO_CC < 0x590)
# define GTEST_DECLARE_TUPLE_AS_FRIEND_ public:
#else
# define GTEST_DECLARE_TUPLE_AS_FRIEND_ \
   template <GTEST_$(n)_TYPENAMES_(U)> friend class tuple; \
  private:
#endif


$range i 0..n-1
$range j 0..n
$range k 1..n
// GTEST_n_TUPLE_(T) is the type of an n-tuple.
#define GTEST_0_TUPLE_(T) tuple<>
```

```
$for k [[
$range m 0..k-1
$range m2 k..n-1
#define GTEST_$(k)_TUPLE_(T) tuple<$for m, [[T##$m]]$for m2 [[, void]]>

]]

// GTEST_n_TYPENAMES_(T) declares a list of n typenames.

$for j [[
$range m 0..j-1
#define GTEST_$(j)_TYPENAMES_(T) $for m, [[typename T##$m]]


]]

// In
 theory, defining stuff in the ::std namespace is undefined
// behavior.  We can do this as we are playing the role of a standard
// library vendor.
namespace std {
namespace tr1 {

template <$for i, [[typename T$i = void]]>
class tuple;

// Anything in namespace gtest_internal is Google Test's INTERNAL
// IMPLEMENTATION DETAIL and MUST NOT BE USED DIRECTLY in user code.
namespace gtest_internal {

// ByRef<T>::type is T if T is a reference; otherwise it's const T&.
template <typename T>
struct ByRef { typedef const T& type; };  // NOLINT
template <typename T>
struct ByRef<T&> { typedef T& type; };  // NOLINT

// A handy wrapper for ByRef.
#define GTEST_BY_REF_(T) typename ::std::tr1::gtest_internal::ByRef<T>::type

// AddRef<T>::type is T if T is a reference; otherwise it's T&.  This
// is the same as tr1::add_reference<T>::type.
template <typename T>
struct AddRef { typedef T& type; };  // NOLINT
template <typename T>
struct AddRef<T&> { typedef T& type; };  // NOLINT

// A handy wrapper
```

```
 for AddRef.
#define GTEST_ADD_REF_(T) typename ::std::tr1::gtest_internal::AddRef<T>::type

// A helper for implementing get<k>().
template <int k> class Get;

// A helper for implementing tuple_element<k, T>.  kIndexValid is true
// iff k < the number of fields in tuple type T.
template <bool kIndexValid, int kIndex, class Tuple>
struct TupleElement;


$for i [[
template <GTEST_$(n)_TYPENAMES_(T)>
struct TupleElement<true, $i, GTEST_$(n)_TUPLE_(T) > {
 typedef T$i type;
};


]]
}  // namespace gtest_internal

template <>
class tuple<> {
public:
 tuple() {}
 tuple(const tuple& /* t */)  {}
 tuple& operator=(const tuple& /* t */) { return *this; }
};


$for k [[
$range m 0..k-1
template <GTEST_$(k)_TYPENAMES_(T)>
class $if k < n [[GTEST_$(k)_TUPLE_(T)]] $else [[tuple]] {
public:
 template <int k> friend class gtest_internal::Get;

 tuple() : $for m, [[f$(m)_()]] {}

 explicit tuple($for m, [[GTEST_BY_REF_(T$m) f$m]]) : [[]]
$for m, [[f$(m)_(f$m)]] {}

 tuple(const tuple& t)
 : $for m, [[f$(m)_(t.f$(m)_)]] {}

 template <GTEST_$(k)_TYPENAMES_(U)>
 tuple(const GTEST_$(k)_TUPLE_(U)& t) : $for m, [[f$(m)_(t.f$(m)_)]] {}
```

```
$if k == 2 [[
 template <typename U0, typename U1>
 tuple(const ::std::pair<U0, U1>& p) : f0_(p.first), f1_(p.second) {}

]]

 tuple& operator=(const tuple& t) { return CopyFrom(t); }

 template <GTEST_$(k)_TYPENAMES_(U)>
 tuple& operator=(const GTEST_$(k)_TUPLE_(U)& t) {
   return CopyFrom(t);
 }

$if k == 2 [[
 template <typename U0, typename U1>
 tuple& operator=(const ::std::pair<U0, U1>& p) {
   f0_ = p.first;
   f1_ = p.second;
   return *this;
 }

]]

 GTEST_DECLARE_TUPLE_AS_FRIEND_

 template <GTEST_$(k)_TYPENAMES_(U)>
 tuple& CopyFrom(const GTEST_$(k)_TUPLE_(U)& t) {

$for m [[
   f$(m)_ = t.f$(m)_;

]]
   return *this;
 }


$for m [[
 T$m f$(m)_;

]]
 };


]]
// 6.1.3.2 Tuple creation functions.

// Known limitations: we don't support passing an
```

```
// std::tr1::reference_wrapper<T> to make_tuple().  And we don't
//
 implement tie().

inline tuple<> make_tuple() { return tuple<>(); }

$for k [[
$range m 0..k-1

template <GTEST_$(k)_TYPENAMES_(T)>
inline GTEST_$(k)_TUPLE_(T) make_tuple($for m, [[const T$m& f$m]]) {
 return GTEST_$(k)_TUPLE_(T)($for m, [[f$m]]);
}

]]

// 6.1.3.3 Tuple helper classes.

template <typename Tuple> struct tuple_size;


$for j [[
template <GTEST_$(j)_TYPENAMES_(T)>
struct tuple_size<GTEST_$(j)_TUPLE_(T) > {
 static const int value = $j;
};


]]
template <int k, class Tuple>
struct tuple_element {
 typedef typename gtest_internal::TupleElement<
     k < (tuple_size<Tuple>::value), k, Tuple>::type type;
};

#define GTEST_TUPLE_ELEMENT_(k, Tuple) typename tuple_element<k, Tuple >::type

// 6.1.3.4 Element access.

namespace gtest_internal {


$for i [[
template <>
class Get<$i> {
public:
 template <class Tuple>
 static GTEST_ADD_REF_(GTEST_TUPLE_ELEMENT_($i, Tuple))
```

```
 Field(Tuple& t) { return t.f$(i)_; }  // NOLINT

 template <class Tuple>
 static GTEST_BY_REF_(GTEST_TUPLE_ELEMENT_($i,
 Tuple))
 ConstField(const Tuple& t) { return t.f$(i)_; }
};


]]
}  // namespace gtest_internal

template <int k, GTEST_$(n)_TYPENAMES_(T)>
GTEST_ADD_REF_(GTEST_TUPLE_ELEMENT_(k, GTEST_$(n)_TUPLE_(T)))
get(GTEST_$(n)_TUPLE_(T)& t) {
 return gtest_internal::Get<k>::Field(t);
}

template <int k, GTEST_$(n)_TYPENAMES_(T)>
GTEST_BY_REF_(GTEST_TUPLE_ELEMENT_(k,  GTEST_$(n)_TUPLE_(T)))
get(const GTEST_$(n)_TUPLE_(T)& t) {
 return gtest_internal::Get<k>::ConstField(t);
}

// 6.1.3.5 Relational operators

// We only implement == and !=, as we don't have a need for the rest yet.

namespace gtest_internal {

// SameSizeTuplePrefixComparator<k, k>::Eq(t1, t2) returns true if the
// first k fields of t1 equals the first k fields of t2.
// SameSizeTuplePrefixComparator(k1, k2) would be a compiler error if
// k1 != k2.
template <int kSize1, int kSize2>
struct SameSizeTuplePrefixComparator;

template <>
struct SameSizeTuplePrefixComparator<0, 0> {
 template <class Tuple1, class
 Tuple2>
 static bool Eq(const Tuple1& /* t1 */, const Tuple2& /* t2 */) {
   return true;
 }
};

template <int k>
struct SameSizeTuplePrefixComparator<k, k> {
```

```
 template <class Tuple1, class Tuple2>
 static bool Eq(const Tuple1& t1, const Tuple2& t2) {
   return SameSizeTuplePrefixComparator<k - 1, k - 1>::Eq(t1, t2) &&
       ::std::tr1::get<k - 1>(t1) == ::std::tr1::get<k - 1>(t2);
 }
};

}  // namespace gtest_internal

template <GTEST_$(n)_TYPENAMES_(T), GTEST_$(n)_TYPENAMES_(U)>
inline bool operator==(const GTEST_$(n)_TUPLE_(T)& t,
                       const GTEST_$(n)_TUPLE_(U)& u) {
 return gtest_internal::SameSizeTuplePrefixComparator<
     tuple_size<GTEST_$(n)_TUPLE_(T) >::value,
     tuple_size<GTEST_$(n)_TUPLE_(U) >::value>::Eq(t, u);
}

template <GTEST_$(n)_TYPENAMES_(T), GTEST_$(n)_TYPENAMES_(U)>
inline bool operator!=(const GTEST_$(n)_TUPLE_(T)& t,
                       const GTEST_$(n)_TUPLE_(U)& u) { return !(t == u); }

// 6.1.4 Pairs.
// Unimplemented.

}
  // namespace tr1
}  // namespace std


$for j [[
#undef GTEST_$(j)_TUPLE_

]]


$for j [[
#undef GTEST_$(j)_TYPENAMES_

]]

#undef GTEST_DECLARE_TUPLE_AS_FRIEND_
#undef GTEST_BY_REF_
#undef GTEST_ADD_REF_
#undef GTEST_TUPLE_ELEMENT_

#endif  // GTEST_INCLUDE_GTEST_INTERNAL_GTEST_TUPLE_H_
```

Found in path(s):

* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-tuple.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-tuple.h.pump
No license file was found, but licenses were detected in source scan.

```
$$ -*- mode: c++; -*-
$var n = 50  $$ Maximum length of Values arguments we want to support.
$var maxtuple = 10  $$ Maximum number of Combine arguments we want to support.
// 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.
//
// Authors: vladl@google.com (Vlad Losev)
//
// Macros and functions for implementing parameterized tests
// in Google C++ Testing Framework (Google Test)
//
// This file is generated by a SCRIPT.  DO
 NOT EDIT BY HAND!
//
#ifndef GTEST_INCLUDE_GTEST_GTEST_PARAM_TEST_H_
#define GTEST_INCLUDE_GTEST_GTEST_PARAM_TEST_H_
```

```
// Value-parameterized tests allow you to test your code with different
// parameters without writing multiple copies of the same test.
//
// Here is how you use value-parameterized tests:

#if 0

// To write value-parameterized tests, first you should define a fixture
// class. It is usually derived from testing::TestWithParam<T> (see below for
// another inheritance scheme that's sometimes useful in more complicated
// class hierarchies), where the type of your parameter values.
// TestWithParam<T> is itself derived from testing::Test. T can be any
// copyable type. If it's a raw pointer, you are responsible for managing the
// lifespan of the pointed values.

class FooTest : public ::testing::TestWithParam<const char*> {
 // You can implement all the usual class fixture members here.
};

// Then, use the TEST_P macro to define as many parameterized tests
// for this
 fixture as you want. The _P suffix is for "parameterized"
// or "pattern", whichever you prefer to think.

TEST_P(FooTest, DoesBlah) {
 // Inside a test, access the test parameter with the GetParam() method
 // of the TestWithParam<T> class:
 EXPECT_TRUE(foo.Blah(GetParam()));
 ...
}

TEST_P(FooTest, HasBlahBlah) {
 ...
}

// Finally, you can use INSTANTIATE_TEST_CASE_P to instantiate the test
// case with any set of parameters you want. Google Test defines a number
// of functions for generating test parameters. They return what we call
// (surprise!) parameter generators. Here is a  summary of them, which
// are all in the testing namespace:
//
//
//  Range(begin, end [, step]) - Yields values {begin, begin+step,
//                               begin+step+step, ...}. The values do not
//                               include end. step defaults to 1.
//  Values(v1, v2, ..., vN)    - Yields values {v1, v2, ..., vN}.
//  ValuesIn(container)        - Yields values from a C-style
```

```
 array, an STL
//  ValuesIn(begin,end)         container, or an iterator range [begin, end).
//  Bool()                      - Yields sequence {false, true}.
//  Combine(g1, g2, ..., gN)    - Yields all combinations (the Cartesian product
//                                for the math savvy) of the values generated
//                                by the N generators.
//
// For more details, see comments at the definitions of these functions below
// in this file.
//
// The following statement will instantiate tests from the FooTest test case
// each with parameter values "meeny", "miny", and "moe".

INSTANTIATE_TEST_CASE_P(InstantiationName,
                        FooTest,
                        Values("meeny", "miny", "moe"));

// To distinguish different instances of the pattern, (yes, you
// can instantiate it more then once) the first argument to the
// INSTANTIATE_TEST_CASE_P macro is a prefix that will be added to the
// actual test case name. Remember to pick unique prefixes for
 different
// instantiations. The tests from the instantiation above will have
// these names:
//
//    * InstantiationName/FooTest.DoesBlah/0 for "meeny"
//    * InstantiationName/FooTest.DoesBlah/1 for "miny"
//    * InstantiationName/FooTest.DoesBlah/2 for "moe"
//    * InstantiationName/FooTest.HasBlahBlah/0 for "meeny"
//    * InstantiationName/FooTest.HasBlahBlah/1 for "miny"
//    * InstantiationName/FooTest.HasBlahBlah/2 for "moe"
//
// You can use these names in --gtest_filter.
//
// This statement will instantiate all tests from FooTest again, each
// with parameter values "cat" and "dog":

const char* pets[] = {"cat", "dog"};
INSTANTIATE_TEST_CASE_P(AnotherInstantiationName, FooTest, ValuesIn(pets));

// The tests from the instantiation above will have these names:
//
//    * AnotherInstantiationName/FooTest.DoesBlah/0 for "cat"
//    * AnotherInstantiationName/FooTest.DoesBlah/1 for "dog"
//    * AnotherInstantiationName/FooTest.HasBlahBlah/0 for "cat"
//    * AnotherInstantiationName/FooTest.HasBlahBlah/1
 for "dog"
//
```

```
// Please note that INSTANTIATE_TEST_CASE_P will instantiate all tests
// in the given test case, whether their definitions come before or
// AFTER the INSTANTIATE_TEST_CASE_P statement.
//
// Please also note that generator expressions (including parameters to the
// generators) are evaluated in InitGoogleTest(), after main() has started.
// This allows the user on one hand, to adjust generator parameters in order
// to dynamically determine a set of tests to run and on the other hand,
// give the user a chance to inspect the generated tests with Google Test
// reflection API before RUN_ALL_TESTS() is executed.
//
// You can see samples/sample7_unittest.cc and samples/sample8_unittest.cc
// for more examples.
//
// In the future, we plan to publish the API for defining new parameter
// generators. But for now this interface remains part of the internal
// implementation and is subject to change.
//
//
// A parameterized test fixture must
 be derived from testing::Test and from
// testing::WithParamInterface<T>, where T is the type of the parameter
// values. Inheriting from TestWithParam<T> satisfies that requirement because
// TestWithParam<T> inherits from both Test and WithParamInterface. In more
// complicated hierarchies, however, it is occasionally useful to inherit
// separately from Test and WithParamInterface. For example:

class BaseTest : public ::testing::Test {
 // You can inherit all the usual members for a non-parameterized test
 // fixture here.
};

class DerivedTest : public BaseTest, public ::testing::WithParamInterface<int> {
 // The usual test fixture members go here too.
};

TEST_F(BaseTest, HasFoo) {
 // This is an ordinary non-parameterized test.
}

TEST_P(DerivedTest, DoesBlah) {
 // GetParam works just the same here as if you inherit from TestWithParam.
 EXPECT_TRUE(foo.Blah(GetParam()));
}

#endif  // 0

#include "gtest/internal/gtest-port.h"
```

```
#if !GTEST_OS_SYMBIAN
# include <utility>
#endif

//
 scripts/fuse_gtest.py depends on gtest's own header being #included
// *unconditionally*.  Therefore these #includes cannot be moved
// inside #if GTEST_HAS_PARAM_TEST.
#include "gtest/internal/gtest-internal.h"
#include "gtest/internal/gtest-param-util.h"
#include "gtest/internal/gtest-param-util-generated.h"

#if GTEST_HAS_PARAM_TEST

namespace testing {

// Functions producing parameter generators.
//
// Google Test uses these generators to produce parameters for value-
// parameterized tests. When a parameterized test case is instantiated
// with a particular generator, Google Test creates and runs tests
// for each element in the sequence produced by the generator.
//
// In the following sample, tests from test case FooTest are instantiated
// each three times with parameter values 3, 5, and 8:
//
// class FooTest : public TestWithParam<int> { ... };
//
// TEST_P(FooTest, TestThis) {
// }
// TEST_P(FooTest, TestThat) {
// }
// INSTANTIATE_TEST_CASE_P(TestSequence, FooTest,
 Values(3, 5, 8));
//

// Range() returns generators providing sequences of values in a range.
//
// Synopsis:
// Range(start, end)
//   - returns a generator producing a sequence of values {start, start+1,
//     start+2, ..., }.
// Range(start, end, step)
//   - returns a generator producing a sequence of values {start, start+step,
//     start+step+step, ..., }.
// Notes:
//   * The generated sequences never include end. For example, Range(1, 5)
```

```
//     returns a generator producing a sequence {1, 2, 3, 4}. Range(1, 9, 2)
//     returns a generator producing {1, 3, 5, 7}.
//   * start and end must have the same type. That type may be any integral or
//     floating-point type or a user defined type satisfying these conditions:
//     * It must be assignable (have operator=() defined).
//     * It must have operator+() (operator+(int-compatible type) for
//       two-operand version).
//     * It must have operator<() defined.
//     Elements in the resulting sequences will also have
 that type.
//   * Condition start < end must be satisfied in order for resulting sequences
//     to contain any elements.
//
template <typename T, typename IncrementT>
internal::ParamGenerator<T> Range(T start, T end, IncrementT step) {
 return internal::ParamGenerator<T>(
    new internal::RangeGenerator<T, IncrementT>(start, end, step));
}

template <typename T>
internal::ParamGenerator<T> Range(T start, T end) {
 return Range(start, end, 1);
}

// ValuesIn() function allows generation of tests with parameters coming from
// a container.
//
// Synopsis:
// ValuesIn(const T (&array)[N])
//   - returns a generator producing sequences with elements from
//     a C-style array.
// ValuesIn(const Container& container)
//   - returns a generator producing sequences with elements from
//     an STL-style container.
// ValuesIn(Iterator begin, Iterator end)
//   - returns a generator producing sequences with elements from
//     a range [begin, end) defined by a pair of STL-style iterators.
 These
//     iterators can also be plain C pointers.
//
// Please note that ValuesIn copies the values from the containers
// passed in and keeps them to generate tests in RUN_ALL_TESTS().
//
// Examples:
//
// This instantiates tests from test case StringTest
// each with C-string values of "foo", "bar", and "baz":
//
```

```
// const char* strings[] = {"foo", "bar", "baz"};
// INSTANTIATE_TEST_CASE_P(StringSequence, SrtingTest, ValuesIn(strings));
//
// This instantiates tests from test case StlStringTest
// each with STL strings with values "a" and "b":
//
// ::std::vector< ::std::string> GetParameterStrings() {
//   ::std::vector< ::std::string> v;
//   v.push_back("a");
//   v.push_back("b");
//   return v;
// }
//
// INSTANTIATE_TEST_CASE_P(CharSequence,
//                         StlStringTest,
//                         ValuesIn(GetParameterStrings()));
//
//
// This will also instantiate tests from CharTest
// each with parameter values 'a' and 'b':
//
// ::std::list<char> GetParameterChars()
 {
//   ::std::list<char> list;
//   list.push_back('a');
//   list.push_back('b');
//   return list;
// }
// ::std::list<char> l = GetParameterChars();
// INSTANTIATE_TEST_CASE_P(CharSequence2,
//                         CharTest,
//                         ValuesIn(l.begin(), l.end()));
//
template <typename ForwardIterator>
internal::ParamGenerator<
 typename ::testing::internal::IteratorTraits<ForwardIterator>::value_type>
ValuesIn(ForwardIterator begin, ForwardIterator end) {
 typedef typename ::testing::internal::IteratorTraits<ForwardIterator>
    ::value_type ParamType;
 return internal::ParamGenerator<ParamType>(
    new internal::ValuesInIteratorRangeGenerator<ParamType>(begin, end));
}

template <typename T, size_t N>
internal::ParamGenerator<T> ValuesIn(const T (&array)[N]) {
 return ValuesIn(array, array + N);
}
```

```
template <class Container>
internal::ParamGenerator<typename Container::value_type> ValuesIn(
  const Container& container) {
 return ValuesIn(container.begin(),
 container.end());
}

// Values() allows generating tests from explicitly specified list of
// parameters.
//
// Synopsis:
// Values(T v1, T v2, ..., T vN)
//   - returns a generator producing sequences with elements v1, v2, ..., vN.
//
// For example, this instantiates tests from test case BarTest each
// with values "one", "two", and "three":
//
// INSTANTIATE_TEST_CASE_P(NumSequence, BarTest, Values("one", "two", "three"));
//
// This instantiates tests from test case BazTest each with values 1, 2, 3.5.
// The exact type of values will depend on the type of parameter in BazTest.
//
// INSTANTIATE_TEST_CASE_P(FloatingNumbers, BazTest, Values(1, 2, 3.5));
//
// Currently, Values() supports from 1 to $n parameters.
//
$range i 1..n
$for i [[
$range j 1..i

template <$for j, [[typename T$j]]>
internal::ValueArray$i<$for j, [[T$j]]> Values($for j, [[T$j v$j]]) {
 return internal::ValueArray$i<$for j, [[T$j]]>($for j, [[v$j]]);
}

]]

// Bool() allows generating
 tests with parameters in a set of (false, true).
//
// Synopsis:
// Bool()
//   - returns a generator producing sequences with elements {false, true}.
//
// It is useful when testing code that depends on Boolean flags. Combinations
// of multiple flags can be tested when several Bool()'s are combined using
// Combine() function.
//
```

```
// In the following example all tests in the test case FlagDependentTest
// will be instantiated twice with parameters false and true.
//
// class FlagDependentTest : public testing::TestWithParam<bool> {
//   virtual void SetUp() {
//     external_flag = GetParam();
//   }
// }
// INSTANTIATE_TEST_CASE_P(BoolSequence, FlagDependentTest, Bool());
//
inline internal::ParamGenerator<bool> Bool() {
 return Values(false, true);
}

# if GTEST_HAS_COMBINE
// Combine() allows the user to combine two or more sequences to produce
// values of a Cartesian product of those sequences' elements.
//
// Synopsis:
// Combine(gen1, gen2, ..., genN)
//   - returns a generator
 producing sequences with elements coming from
//     the Cartesian product of elements from the sequences generated by
//     gen1, gen2, ..., genN. The sequence elements will have a type of
//     tuple<T1, T2, ..., TN> where T1, T2, ..., TN are the types
//     of elements from sequences produces by gen1, gen2, ..., genN.
//
// Combine can have up to $maxtuple arguments. This number is currently limited
// by the maximum number of elements in the tuple implementation used by Google
// Test.
//
// Example:
//
// This will instantiate tests in test case AnimalTest each one with
// the parameter values tuple("cat", BLACK), tuple("cat", WHITE),
// tuple("dog", BLACK), and tuple("dog", WHITE):
//
// enum Color { BLACK, GRAY, WHITE };
// class AnimalTest
//     : public testing::TestWithParam<tuple<const char*, Color> > {...};
//
// TEST_P(AnimalTest, AnimalLooksNice) {...}
//
// INSTANTIATE_TEST_CASE_P(AnimalVariations, AnimalTest,
//                         Combine(Values("cat", "dog"),
//
                                 Values(BLACK, WHITE)));
//
```

```
// This will instantiate tests in FlagDependentTest with all variations of two
// Boolean flags:
//
// class FlagDependentTest
//     : public testing::TestWithParam<tuple<bool, bool> > {
//   virtual void SetUp() {
//     // Assigns external_flag_1 and external_flag_2 values from the tuple.
//     tie(external_flag_1, external_flag_2) = GetParam();
//   }
// };
//
// TEST_P(FlagDependentTest, TestFeature1) {
//   // Test your code using external_flag_1 and external_flag_2 here.
// }
// INSTANTIATE_TEST_CASE_P(TwoBoolSequence, FlagDependentTest,
//                         Combine(Bool(), Bool()));
//
$range i 2..maxtuple
$for i [[
$range j 1..i

template <$for j, [[typename Generator$j]]>
internal::CartesianProductHolder$i<$for j, [[Generator$j]]> Combine(
   $for j, [[const Generator$j& g$j]]) {
 return internal::CartesianProductHolder$i<$for j, [[Generator$j]]>(
    $for j, [[g$j]]);
}

]]
# endif  // GTEST_HAS_COMBINE



#
 define TEST_P(test_case_name, test_name) \
 class GTEST_TEST_CLASS_NAME_(test_case_name, test_name) \
    : public test_case_name { \
  public: \
   GTEST_TEST_CLASS_NAME_(test_case_name, test_name)() {} \
   virtual void TestBody(); \
  private: \
   static int AddToRegistry() { \
     ::testing::UnitTest::GetInstance()->parameterized_test_registry(). \
        GetTestCasePatternHolder<test_case_name>(\
           #test_case_name, __FILE__, __LINE__)->AddTestPattern(\
              #test_case_name, \
              #test_name, \
              new ::testing::internal::TestMetaFactory< \
```

```
                GTEST_TEST_CLASS_NAME_(test_case_name, test_name)>()); \
    return 0; \
  } \
  static int gtest_registering_dummy_; \
  GTEST_DISALLOW_COPY_AND_ASSIGN_(\
      GTEST_TEST_CLASS_NAME_(test_case_name, test_name)); \
 }; \
 int GTEST_TEST_CLASS_NAME_(test_case_name, \
                    test_name)::gtest_registering_dummy_ = \
     GTEST_TEST_CLASS_NAME_(test_case_name, test_name)::AddToRegistry(); \
 void GTEST_TEST_CLASS_NAME_(test_case_name, test_name)::TestBody()

# define INSTANTIATE_TEST_CASE_P(prefix, test_case_name, generator) \
 ::testing::internal::ParamGenerator<test_case_name::ParamType> \
    gtest_##prefix##test_case_name##_EvalGenerator_() { return generator; } \
 int gtest_##prefix##test_case_name##_dummy_ = \
    ::testing::UnitTest::GetInstance()->parameterized_test_registry(). \
        GetTestCasePatternHolder<test_case_name>(\
           #test_case_name, __FILE__, __LINE__)->AddTestCaseInstantiation(\
              #prefix, \
              &gtest_##prefix##test_case_name##_EvalGenerator_, \
              __FILE__, __LINE__)

}  // namespace testing

#endif  // GTEST_HAS_PARAM_TEST

#endif  // GTEST_INCLUDE_GTEST_GTEST_PARAM_TEST_H_
```

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/gtest-param-test.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/gtest-param-test.h.pump
No license file was found, but licenses were detected in source scan.

```
$$ -*- mode: c++; -*-
$$ This is a Pump source file.  Please use Pump to convert it to
$$ gmock-generated-actions.h.
$$
$var n = 10  $$ The maximum arity we support.
$$ }} This line fixes auto-indentation of the following code in Emacs.
// 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.

// Google Mock - a framework for writing C++ mock classes.
//
// This file implements some commonly used variadic matchers.

#ifndef GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_MATCHERS_H_
#define
 GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_MATCHERS_H_

#include <iterator>
#include <sstream>
#include <string>
#include <vector>
#include "gmock/gmock-matchers.h"

namespace testing {
namespace internal {

$range i 0..n-1

// The type of the i-th (0-based) field of Tuple.
#define GMOCK_FIELD_TYPE_(Tuple, i) \
  typename ::std::tr1::tuple_element<i, Tuple>::type

// TupleFields<Tuple, k0, ..., kn> is for selecting fields from a
// tuple of type Tuple.  It has two members:
//
```

```
//   type: a tuple type whose i-th field is the ki-th field of Tuple.
//   GetSelectedFields(t): returns fields k0, ..., and kn of t as a tuple.
//
// For example, in class TupleFields<tuple<bool, char, int>, 2, 0>, we have:
//
//   type is tuple<int, bool>, and
//   GetSelectedFields(make_tuple(true, 'a', 42)) is (42, true).

template <class Tuple$for i [[, int k$i = -1]]>
class TupleFields;

// This generic version is used when there are $n selectors.
template <class
 Tuple$for i [[, int k$i]]>
class TupleFields {
public:
 typedef ::std::tr1::tuple<$for i, [[GMOCK_FIELD_TYPE_(Tuple, k$i)]]> type;
 static type GetSelectedFields(const Tuple& t) {
   using ::std::tr1::get;
   return type($for i, [[get<k$i>(t)]]);
 }
};

// The following specialization is used for 0 ~ $(n-1) selectors.

$for i [[
$$ }}}
$range j 0..i-1
$range k 0..n-1

template <class Tuple$for j [[, int k$j]]>
class TupleFields<Tuple, $for k, [[$if k < i [[k$k]] $else [[-1]]]]> {
public:
 typedef ::std::tr1::tuple<$for j, [[GMOCK_FIELD_TYPE_(Tuple, k$j)]]> type;
 static type GetSelectedFields(const Tuple& $if i==0 [[/* t */]] $else [[t]]) {
   using ::std::tr1::get;
   return type($for j, [[get<k$j>(t)]]);
 }
};

]]

#undef GMOCK_FIELD_TYPE_

// Implements the Args() matcher.

$var ks = [[$for i, [[k$i]]]]
template <class ArgsTuple$for i [[, int k$i = -1]]>
```

```
class ArgsMatcherImpl : public MatcherInterface<ArgsTuple> {
public:
 // ArgsTuple may have top-level const or reference
 modifiers.
 typedef GTEST_REMOVE_REFERENCE_AND_CONST_(ArgsTuple) RawArgsTuple;
 typedef typename internal::TupleFields<RawArgsTuple, $ks>::type SelectedArgs;
 typedef Matcher<const SelectedArgs&> MonomorphicInnerMatcher;

 template <typename InnerMatcher>
 explicit ArgsMatcherImpl(const InnerMatcher& inner_matcher)
    : inner_matcher_(SafeMatcherCast<const SelectedArgs&>(inner_matcher)) {}

 virtual bool MatchAndExplain(ArgsTuple args,
                              MatchResultListener* listener) const {
  const SelectedArgs& selected_args = GetSelectedArgs(args);
  if (!listener->IsInterested())
    return inner_matcher_.Matches(selected_args);

  PrintIndices(listener->stream());
  *listener << "are " << PrintToString(selected_args);

  StringMatchResultListener inner_listener;
  const bool match = inner_matcher_.MatchAndExplain(selected_args,
                                                    &inner_listener);
  PrintIfNotEmpty(inner_listener.str(),
listener->stream());
  return match;
 }

 virtual void DescribeTo(::std::ostream* os) const {
  *os << "are a tuple ";
  PrintIndices(os);
  inner_matcher_.DescribeTo(os);
 }

 virtual void DescribeNegationTo(::std::ostream* os) const {
  *os << "are a tuple ";
  PrintIndices(os);
  inner_matcher_.DescribeNegationTo(os);
 }

private:
 static SelectedArgs GetSelectedArgs(ArgsTuple args) {
  return TupleFields<RawArgsTuple, $ks>::GetSelectedFields(args);
 }

 // Prints the indices of the selected fields.
 static void PrintIndices(::std::ostream* os) {
```

```
  *os << "whose fields (";
  const int indices[$n] = { $ks };
  for (int i = 0; i < $n; i++) {
    if (indices[i] < 0)
      break;

    if (i >= 1)
      *os << ", ";

    *os << "#" << indices[i];
  }
  *os << ") ";
 }

 const MonomorphicInnerMatcher inner_matcher_;

 GTEST_DISALLOW_ASSIGN_(ArgsMatcherImpl);
};

template <class InnerMatcher$for i [[, int k$i = -1]]>
class ArgsMatcher {
 public:
 explicit ArgsMatcher(const InnerMatcher& inner_matcher)
     : inner_matcher_(inner_matcher) {}

 template <typename ArgsTuple>
 operator Matcher<ArgsTuple>() const {
   return MakeMatcher(new ArgsMatcherImpl<ArgsTuple, $ks>(inner_matcher_));
 }

private:
 const InnerMatcher inner_matcher_;

 GTEST_DISALLOW_ASSIGN_(ArgsMatcher);
};

// A set of metafunctions for computing the result type of AllOf.
// AllOf(m1, ..., mN) returns
// AllOfResultN<decltype(m1), ..., decltype(mN)>::type.

// Although AllOf isn't defined for one argument, AllOfResult1 is defined
// to simplify the implementation.
template <typename M1>
struct AllOfResult1 {
 typedef M1 type;
};

$range i 1..n
```

```
$range i 2..n
$for i [[
$range j 2..i
$var m = i/2
$range k 1..m
$range t m+1..i

template <typename M1$for j [[, typename M$j]]>
struct AllOfResult$i {
 typedef BothOfMatcher<
    typename AllOfResult$m<$for k, [[M$k]]>::type,
    typename AllOfResult$(i-m)<$for t, [[M$t]]>::type
 > type;
};

]]

//
 A set of metafunctions for computing the result type of AnyOf.
// AnyOf(m1, ..., mN) returns
// AnyOfResultN<decltype(m1), ..., decltype(mN)>::type.

// Although AnyOf isn't defined for one argument, AnyOfResult1 is defined
// to simplify the implementation.
template <typename M1>
struct AnyOfResult1 {
 typedef M1 type;
};

$range i 1..n

$range i 2..n
$for i [[
$range j 2..i
$var m = i/2
$range k 1..m
$range t m+1..i

template <typename M1$for j [[, typename M$j]]>
struct AnyOfResult$i {
 typedef EitherOfMatcher<
    typename AnyOfResult$m<$for k, [[M$k]]>::type,
    typename AnyOfResult$(i-m)<$for t, [[M$t]]>::type
 > type;
};

]]
```

```
}  // namespace internal

// Args<N1, N2, ..., Nk>(a_matcher) matches a tuple if the selected
// fields of it matches a_matcher.  C++ doesn't support default
// arguments for function templates, so we have to overload it.

$range i 0..n
$for i [[
$range j 1..i
template <$for j [[int k$j, ]]typename InnerMatcher>
inline internal::ArgsMatcher<InnerMatcher$for
 j [[, k$j]]>
Args(const InnerMatcher& matcher) {
 return internal::ArgsMatcher<InnerMatcher$for j [[, k$j]]>(matcher);
}



]]
// ElementsAre(e_1, e_2, ... e_n) matches an STL-style container with
// n elements, where the i-th element in the container must
// match the i-th argument in the list.  Each argument of
// ElementsAre() can be either a value or a matcher.  We support up to
// $n arguments.
//
// The use of DecayArray in the implementation allows ElementsAre()
// to accept string literals, whose type is const char[N], but we
// want to treat them as const char*.
//
// NOTE: Since ElementsAre() cares about the order of the elements, it
// must not be used with containers whose elements's order is
// undefined (e.g. hash_map).

$range i 0..n
$for i [[

$range j 1..i

$if i>0 [[

template <$for j, [[typename T$j]]>
]]

inline internal::ElementsAreMatcher<
  std::tr1::tuple<
$for j, [[

      typename internal::DecayArray<T$j[[]]>::type]]> >
```

```
ElementsAre($for j,
 [[const T$j& e$j]]) {
 typedef std::tr1::tuple<
$for j, [[

    typename internal::DecayArray<T$j[[]]>::type]]> Args;
 return internal::ElementsAreMatcher<Args>(Args($for j, [[e$j]]));
}

]]

// UnorderedElementsAre(e_1, e_2, ..., e_n) is an ElementsAre extension
// that matches n elements in any order.  We support up to n=$n arguments.

$range i 0..n
$for i [[

$range j 1..i

$if i>0 [[

template <$for j, [[typename T$j]]>
]]

inline internal::UnorderedElementsAreMatcher<
  std::tr1::tuple<
$for j, [[

      typename internal::DecayArray<T$j[[]]>::type]]> >
UnorderedElementsAre($for j, [[const T$j& e$j]]) {
 typedef std::tr1::tuple<
$for j, [[

    typename internal::DecayArray<T$j[[]]>::type]]> Args;
 return internal::UnorderedElementsAreMatcher<Args>(Args($for j, [[e$j]]));
}

]]

// AllOf(m1, m2, ..., mk) matches any value that matches all of the given
// sub-matchers.  AllOf is called fully qualified to prevent ADL from firing.

$range i 2..n
$for i [[
$range j 1..i
$var
 m = i/2
$range k 1..m
```

```
$range t m+1..i

template <$for j, [[typename M$j]]>
inline typename internal::AllOfResult$i<$for j, [[M$j]]>::type
AllOf($for j, [[M$j m$j]]) {
 return typename internal::AllOfResult$i<$for j, [[M$j]]>::type(
    $if m == 1 [[m1]] $else [[::testing::AllOf($for k, [[m$k]])]],
    $if m+1 == i [[m$i]] $else [[::testing::AllOf($for t, [[m$t]])]]);
}

]]

// AnyOf(m1, m2, ..., mk) matches any value that matches any of the given
// sub-matchers.  AnyOf is called fully qualified to prevent ADL from firing.

$range i 2..n
$for i [[
$range j 1..i
$var m = i/2
$range k 1..m
$range t m+1..i

template <$for j, [[typename M$j]]>
inline typename internal::AnyOfResult$i<$for j, [[M$j]]>::type
AnyOf($for j, [[M$j m$j]]) {
 return typename internal::AnyOfResult$i<$for j, [[M$j]]>::type(
    $if m == 1 [[m1]] $else [[::testing::AnyOf($for k, [[m$k]])]],
    $if m+1 == i [[m$i]] $else [[::testing::AnyOf($for t, [[m$t]])]]);
}

]]

}  // namespace testing
$$ } // This
 Pump meta comment fixes auto-indentation in Emacs. It will not
$$   // show up in the generated code.


// The MATCHER* family of macros can be used in a namespace scope to
// define custom matchers easily.
//
// Basic Usage
// ===========
//
// The syntax
//
//   MATCHER(name, description_string) { statements; }
//
```

```
// defines a matcher with the given name that executes the statements,
// which must return a bool to indicate if the match succeeds.  Inside
// the statements, you can refer to the value being matched by 'arg',
// and refer to its type by 'arg_type'.
//
// The description string documents what the matcher does, and is used
// to generate the failure message when the match fails.  Since a
// MATCHER() is usually defined in a header file shared by multiple
// C++ source files, we require the description to be a C-string
// literal to avoid possible side effects.  It can be empty, in which
// case we'll use the sequence of words in the matcher name as the
// description.
//
//
 For example:
//
//   MATCHER(IsEven, "") { return (arg % 2) == 0; }
//
// allows you to write
//
//   // Expects mock_foo.Bar(n) to be called where n is even.
//   EXPECT_CALL(mock_foo, Bar(IsEven()));
//
// or,
//
//   // Verifies that the value of some_expression is even.
//   EXPECT_THAT(some_expression, IsEven());
//
// If the above assertion fails, it will print something like:
//
//   Value of: some_expression
//   Expected: is even
//     Actual: 7
//
// where the description "is even" is automatically calculated from the
// matcher name IsEven.
//
// Argument Type
// =============
//
// Note that the type of the value being matched (arg_type) is
// determined by the context in which you use the matcher and is
// supplied to you by the compiler, so you don't need to worry about
// declaring it (nor can you).  This allows the matcher to be
// polymorphic.  For example, IsEven() can be used to match any type
// where the value of "(arg % 2) == 0" can be implicitly converted to
//
```

```
 a bool.  In the "Bar(IsEven())" example above, if method Bar()
// takes an int, 'arg_type' will be int; if it takes an unsigned long,
// 'arg_type' will be unsigned long; and so on.
//
// Parameterizing Matchers
// =======================
//
// Sometimes you'll want to parameterize the matcher.  For that you
// can use another macro:
//
//   MATCHER_P(name, param_name, description_string) { statements; }
//
// For example:
//
//   MATCHER_P(HasAbsoluteValue, value, "") { return abs(arg) == value; }
//
// will allow you to write:
//
//   EXPECT_THAT(Blah("a"), HasAbsoluteValue(n));
//
// which may lead to this message (assuming n is 10):
//
//   Value of: Blah("a")
//   Expected: has absolute value 10
//     Actual: -9
//
// Note that both the matcher description and its parameter are
// printed, making the message human-friendly.
//
// In the matcher definition body, you can write 'foo_type' to
// reference the type of a parameter named 'foo'.  For example, in the
// body of MATCHER_P(HasAbsoluteValue,
 value) above, you can write
// 'value_type' to refer to the type of 'value'.
//
// We also provide MATCHER_P2, MATCHER_P3, ..., up to MATCHER_P$n to
// support multi-parameter matchers.
//
// Describing Parameterized Matchers
// =================================
//
// The last argument to MATCHER*() is a string-typed expression.  The
// expression can reference all of the matcher's parameters and a
// special bool-typed variable named 'negation'.  When 'negation' is
// false, the expression should evaluate to the matcher's description;
// otherwise it should evaluate to the description of the negation of
// the matcher.  For example,
//
```

```
//   using testing::PrintToString;
//
//   MATCHER_P2(InClosedRange, low, hi,
//       string(negation ? "is not" : "is") + " in range [" +
//       PrintToString(low) + ", " + PrintToString(hi) + "]") {
//     return low <= arg && arg <= hi;
//   }
//   ...
//   EXPECT_THAT(3, InClosedRange(4, 6));
//   EXPECT_THAT(3, Not(InClosedRange(2,
 4)));
//
// would generate two failures that contain the text:
//
//   Expected: is in range [4, 6]
//   ...
//   Expected: is not in range [2, 4]
//
// If you specify "" as the description, the failure message will
// contain the sequence of words in the matcher name followed by the
// parameter values printed as a tuple.  For example,
//
//   MATCHER_P2(InClosedRange, low, hi, "") { ... }
//   ...
//   EXPECT_THAT(3, InClosedRange(4, 6));
//   EXPECT_THAT(3, Not(InClosedRange(2, 4)));
//
// would generate two failures that contain the text:
//
//   Expected: in closed range (4, 6)
//   ...
//   Expected: not (in closed range (2, 4))
//
// Types of Matcher Parameters
// ===========================
//
// For the purpose of typing, you can view
//
//   MATCHER_Pk(Foo, p1, ..., pk, description_string) { ... }
//
// as shorthand for
//
//   template <typename p1_type, ..., typename pk_type>
//   FooMatcherPk<p1_type, ..., pk_type>
//   Foo(p1_type p1, ..., pk_type pk) { ... }
//
//
```

```
 When you write Foo(v1, ..., vk), the compiler infers the types of
// the parameters v1, ..., and vk for you.  If you are not happy with
// the result of the type inference, you can specify the types by
// explicitly instantiating the template, as in Foo<long, bool>(5,
// false).  As said earlier, you don't get to (or need to) specify
// 'arg_type' as that's determined by the context in which the matcher
// is used.  You can assign the result of expression Foo(p1, ..., pk)
// to a variable of type FooMatcherPk<p1_type, ..., pk_type>.  This
// can be useful when composing matchers.
//
// While you can instantiate a matcher template with reference types,
// passing the parameters by pointer usually makes your code more
// readable.  If, however, you still want to pass a parameter by
// reference, be aware that in the failure message generated by the
// matcher you will see the value of the referenced object but not its
// address.
//
// Explaining Match Results
// ========================
//
//
 Sometimes the matcher description alone isn't enough to explain why
// the match has failed or succeeded.  For example, when expecting a
// long string, it can be very helpful to also print the diff between
// the expected string and the actual one.  To achieve that, you can
// optionally stream additional information to a special variable
// named result_listener, whose type is a pointer to class
// MatchResultListener:
//
//   MATCHER_P(EqualsLongString, str, "") {
//     if (arg == str) return true;
//
//     *result_listener << "the difference: "
///                      << DiffStrings(str, arg);
//     return false;
//   }
//
// Overloading Matchers
// ====================
//
// You can overload matchers with different numbers of parameters:
//
//   MATCHER_P(Blah, a, description_string1) { ... }
//   MATCHER_P2(Blah, a, b, description_string2) { ... }
//
// Caveats
// =======
//
```

```
// When defining a new matcher, you should also consider implementing
// MatcherInterface or using
 MakePolymorphicMatcher().  These
// approaches require more work than the MATCHER* macros, but also
// give you more control on the types of the value being matched and
// the matcher parameters, which may leads to better compiler error
// messages when the matcher is used wrong.  They also allow
// overloading matchers based on parameter types (as opposed to just
// based on the number of parameters).
//
// MATCHER*() can only be used in a namespace scope.  The reason is
// that C++ doesn't yet allow function-local types to be used to
// instantiate templates.  The up-coming C++0x standard will fix this.
// Once that's done, we'll consider supporting using MATCHER*() inside
// a function.
//
// More Information
// ================
//
// To learn more about using these macros, please search for 'MATCHER'
// on http://code.google.com/p/googlemock/wiki/CookBook.

$range i 0..n
$for i

[[
$var macro_name = [[$if i==0 [[MATCHER]] $elif i==1 [[MATCHER_P]]

          $else [[MATCHER_P$i]]]]
$var class_name = [[name##Matcher[[$if i==0 [[]] $elif i==1 [[P]]
                                      $else [[P$i]]]]]]
$range j 0..i-1
$var template = [[$if i==0 [[]] $else [[

 template <$for j, [[typename p$j##_type]]>\
]]]]
$var ctor_param_list = [[$for j, [[p$j##_type gmock_p$j]]]]
$var impl_ctor_param_list = [[$for j, [[p$j##_type gmock_p$j]]]]
$var impl_inits = [[$if i==0 [[]] $else [[ : $for j, [[p$j(gmock_p$j)]]]]]]
$var inits = [[$if i==0 [[]] $else [[ : $for j, [[p$j(gmock_p$j)]]]]]]
$var params = [[$for j, [[p$j]]]]
$var param_types = [[$if i==0 [[]] $else [[<$for j, [[p$j##_type]]>]]]]
$var param_types_and_names = [[$for j, [[p$j##_type p$j]]]]
$var param_field_decls = [[$for j
[[

    p$j##_type p$j;\
]]]]
```

```
$var param_field_decls2 = [[$for j
[[

  p$j##_type p$j;\
]]]]

#define $macro_name(name$for j [[, p$j]], description)\$template
 class $class_name {\
  public:\
   template <typename arg_type>\
   class gmock_Impl
 : public ::testing::MatcherInterface<arg_type> {\
    public:\
     [[$if i==1 [[explicit ]]]]gmock_Impl($impl_ctor_param_list)\
         $impl_inits {}\
     virtual bool MatchAndExplain(\
         arg_type arg, ::testing::MatchResultListener* result_listener) const;\
     virtual void DescribeTo(::std::ostream* gmock_os) const {\
       *gmock_os << FormatDescription(false);\
     }\
     virtual void DescribeNegationTo(::std::ostream* gmock_os) const {\
       *gmock_os << FormatDescription(true);\
     }\$param_field_decls
    private:\
     ::testing::internal::string FormatDescription(bool negation) const {\
       const ::testing::internal::string gmock_description = (description);\
       if (!gmock_description.empty())\
         return gmock_description;\
       return ::testing::internal::FormatMatcherDescription(\
           negation, #name, \
           ::testing::internal::UniversalTersePrintTupleFieldsToStrings(\
               ::std::tr1::tuple<$for
 j, [[p$j##_type]]>($for j, [[p$j]])));\
     }\
     GTEST_DISALLOW_ASSIGN_(gmock_Impl);\
   };\
   template <typename arg_type>\
   operator ::testing::Matcher<arg_type>() const {\
     return ::testing::Matcher<arg_type>(\
         new gmock_Impl<arg_type>($params));\
   }\
   $class_name($ctor_param_list)$inits {\
   }\$param_field_decls2
  private:\
   GTEST_DISALLOW_ASSIGN_($class_name);\
 };\$template
 inline $class_name$param_types name($param_types_and_names) {\
   return $class_name$param_types($params);\
```

```
 }\$template
 template <typename arg_type>\
 bool $class_name$param_types::gmock_Impl<arg_type>::MatchAndExplain(\
    arg_type arg, \
    ::testing::MatchResultListener* result_listener GTEST_ATTRIBUTE_UNUSED_)\
       const
]]


#endif  // GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_MATCHERS_H_
```

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-generated-matchers.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-generated-matchers.h.pump
No license file was found, but licenses were detected in source scan.

```
$$ -*- mode: c++; -*-
$var n = 50  $$ Maximum length of Values arguments we want to support.
$var maxtuple = 10  $$ Maximum number of Combine arguments we want to support.
// 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.
//
// Author: vladl@google.com (Vlad Losev)

// Type and function utilities for implementing parameterized tests.
// This file is generated by a SCRIPT.  DO NOT EDIT BY HAND!
//
// Currently Google Test
 supports at most $n arguments in Values,
// and at most $maxtuple arguments in Combine. Please contact
// googletestframework@googlegroups.com if you need more.
// Please note that the number of arguments to Combine is limited
// by the maximum arity of the implementation of tr1::tuple which is
// currently set at $maxtuple.

#ifndef GTEST_INCLUDE_GTEST_INTERNAL_GTEST_PARAM_UTIL_GENERATED_H_
#define GTEST_INCLUDE_GTEST_INTERNAL_GTEST_PARAM_UTIL_GENERATED_H_

// scripts/fuse_gtest.py depends on gtest's own header being #included
// *unconditionally*.  Therefore these #includes cannot be moved
// inside #if GTEST_HAS_PARAM_TEST.
#include "gtest/internal/gtest-param-util.h"
#include "gtest/internal/gtest-port.h"

#if GTEST_HAS_PARAM_TEST

namespace testing {

// Forward declarations of ValuesIn(), which is implemented in
// include/gtest/gtest-param-test.h.
template <typename ForwardIterator>
internal::ParamGenerator<
 typename ::testing::internal::IteratorTraits<ForwardIterator>::value_type>
ValuesIn(ForwardIterator
 begin, ForwardIterator end);

template <typename T, size_t N>
internal::ParamGenerator<T> ValuesIn(const T (&array)[N]);

template <class Container>
internal::ParamGenerator<typename Container::value_type> ValuesIn(
   const Container& container);

namespace internal {

// Used in the Values() function to provide polymorphic capabilities.
template <typename T1>
class ValueArray1 {
```

```
public:
 explicit ValueArray1(T1 v1) : v1_(v1) {}

 template <typename T>
 operator ParamGenerator<T>() const { return ValuesIn(&v1_, &v1_ + 1); }

private:
 // No implementation - assignment is unsupported.
 void operator=(const ValueArray1& other);

 const T1 v1_;
};

$range i 2..n
$for i [[
$range j 1..i

template <$for j, [[typename T$j]]>
class ValueArray$i {
public:
 ValueArray$i($for j, [[T$j v$j]]) : $for j, [[v$(j)_(v$j)]] {}

 template <typename T>
 operator ParamGenerator<T>() const {
   const T array[] = {$for j, [[static_cast<T>(v$(j)_)]]};
   return
 ValuesIn(array);
 }

private:
 // No implementation - assignment is unsupported.
 void operator=(const ValueArray$i& other);

$for j [[

 const T$j v$(j)_;
]]

};

]]

# if GTEST_HAS_COMBINE
// INTERNAL IMPLEMENTATION - DO NOT USE IN USER CODE.
//
// Generates values from the Cartesian product of values produced
// by the argument generators.
//
```

```
$range i 2..maxtuple
$for i [[
$range j 1..i
$range k 2..i

template <$for j, [[typename T$j]]>
class CartesianProductGenerator$i
  : public ParamGeneratorInterface< ::std::tr1::tuple<$for j, [[T$j]]> > {
public:
 typedef ::std::tr1::tuple<$for j, [[T$j]]> ParamType;

 CartesianProductGenerator$i($for j, [[const ParamGenerator<T$j>& g$j]])
    : $for j, [[g$(j)_(g$j)]] {}
 virtual ~CartesianProductGenerator$i() {}

 virtual ParamIteratorInterface<ParamType>* Begin() const {
  return new Iterator(this, $for j, [[g$(j)_, g$(j)_.begin()]]);
 }
 virtual ParamIteratorInterface<ParamType>* End() const {
  return new Iterator(this,
 $for j, [[g$(j)_, g$(j)_.end()]]);
 }

private:
 class Iterator : public ParamIteratorInterface<ParamType> {
  public:
   Iterator(const ParamGeneratorInterface<ParamType>* base, $for j, [[

     const ParamGenerator<T$j>& g$j,
     const typename ParamGenerator<T$j>::iterator& current$(j)]])
       : base_(base),
$for j, [[

         begin$(j)_(g$j.begin()), end$(j)_(g$j.end()), current$(j)_(current$j)
]]    {
     ComputeCurrentValue();
   }
   virtual ~Iterator() {}

   virtual const ParamGeneratorInterface<ParamType>* BaseGenerator() const {
    return base_;
   }
   // Advance should not be called on beyond-of-range iterators
   // so no component iterators must be beyond end of range, either.
   virtual void Advance() {
    assert(!AtEnd());
    ++current$(i)_;
```

```
$for k [[
    if (current$(i+2-k)_ == end$(i+2-k)_) {
      current$(i+2-k)_ = begin$(i+2-k)_;
      ++current$(i+2-k-1)_;
    }

]]
    ComputeCurrentValue();
  }
  virtual ParamIteratorInterface<ParamType>*
 Clone() const {
    return new Iterator(*this);
  }
  virtual const ParamType* Current() const { return &current_value_; }
  virtual bool Equals(const ParamIteratorInterface<ParamType>& other) const {
    // Having the same base generator guarantees that the other
    // iterator is of the same type and we can downcast.
    GTEST_CHECK_(BaseGenerator() == other.BaseGenerator())
        << "The program attempted to compare iterators "
        << "from different generators." << std::endl;
    const Iterator* typed_other =
        CheckedDowncastToActualType<const Iterator>(&other);
    // We must report iterators equal if they both point beyond their
    // respective ranges. That can happen in a variety of fashions,
    // so we have to consult AtEnd().
    return (AtEnd() && typed_other->AtEnd()) ||
       ($for j  && [[

        current$(j)_ == typed_other->current$(j)_
]]);
  }

 private:
  Iterator(const Iterator& other)
      : base_(other.base_), $for j, [[

      begin$(j)_(other.begin$(j)_),
      end$(j)_(other.end$(j)_),
      current$(j)_(other.current$(j)_)
]] {
    ComputeCurrentValue();
  }

  void ComputeCurrentValue() {
    if (!AtEnd())
      current_value_ = ParamType($for j, [[*current$(j)_]]);
  }
  bool AtEnd() const {
```

```
    // We must report iterator past the end of the range when either of the
    // component iterators has reached the end of its range.
    return
$for j  || [[

        current$(j)_ == end$(j)_
]];
  }

  // No implementation - assignment is unsupported.
  void operator=(const Iterator& other);

  const ParamGeneratorInterface<ParamType>* const base_;
  // begin[i]_ and end[i]_ define the i-th range that Iterator traverses.
  // current[i]_ is the actual traversing iterator.
$for j [[

  const typename ParamGenerator<T$j>::iterator begin$(j)_;
  const typename ParamGenerator<T$j>::iterator end$(j)_;
  typename ParamGenerator<T$j>::iterator
 current$(j)_;
]]

  ParamType current_value_;
 };  // class CartesianProductGenerator$i::Iterator

 // No implementation - assignment is unsupported.
 void operator=(const CartesianProductGenerator$i& other);


$for j [[
 const ParamGenerator<T$j> g$(j)_;

]]
};  // class CartesianProductGenerator$i



]]

// INTERNAL IMPLEMENTATION - DO NOT USE IN USER CODE.
//
// Helper classes providing Combine() with polymorphic features. They allow
// casting CartesianProductGeneratorN<T> to ParamGenerator<U> if T is
// convertible to U.
//
$range i 2..maxtuple
$for i [[
$range j 1..i
```

```
template <$for j, [[class Generator$j]]>
class CartesianProductHolder$i {
public:
CartesianProductHolder$i($for j, [[const Generator$j& g$j]])
    : $for j, [[g$(j)_(g$j)]] {}
 template <$for j, [[typename T$j]]>
 operator ParamGenerator< ::std::tr1::tuple<$for j, [[T$j]]> >() const {
   return ParamGenerator< ::std::tr1::tuple<$for j, [[T$j]]> >(
       new CartesianProductGenerator$i<$for
 j, [[T$j]]>(
$for j,[[

       static_cast<ParamGenerator<T$j> >(g$(j)_)
]]));
 }

private:
 // No implementation - assignment is unsupported.
 void operator=(const CartesianProductHolder$i& other);


$for j [[
 const Generator$j g$(j)_;

]]
};  // class CartesianProductHolder$i

]]

# endif  // GTEST_HAS_COMBINE

}  // namespace internal
}  // namespace testing

#endif  //  GTEST_HAS_PARAM_TEST

#endif  // GTEST_INCLUDE_GTEST_INTERNAL_GTEST_PARAM_UTIL_GENERATED_H_
```

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gtest-1.7.0/include/gtest/internal/gtest-param-util-generated.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/gtest/internal/gtest-param-util-generated.h.pump
No license file was found, but licenses were detected in source scan.

```
$$ -*- mode: c++; -*-
$$ This is a Pump source file.  Please use Pump to convert it to
$$ gmock-generated-nice-strict.h.
```

```
$$
$var n = 10  $$ The maximum arity we support.
// 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.
//
// Author: wan@google.com (Zhanyong Wan)

// Implements class templates NiceMock, NaggyMock, and StrictMock.
//
// Given a mock class MockFoo that is created using Google Mock,
// NiceMock<MockFoo> is
 a subclass of MockFoo that allows
// uninteresting calls (i.e. calls to mock methods that have no
// EXPECT_CALL specs), NaggyMock<MockFoo> is a subclass of MockFoo
// that prints a warning when an uninteresting call occurs, and
// StrictMock<MockFoo> is a subclass of MockFoo that treats all
// uninteresting calls as errors.
//
// Currently a mock is naggy by default, so MockFoo and
// NaggyMock<MockFoo> behave like the same.  However, we will soon
// switch the default behavior of mocks to be nice, as that in general
```

```
// leads to more maintainable tests.  When that happens, MockFoo will
// stop behaving like NaggyMock<MockFoo> and start behaving like
// NiceMock<MockFoo>.
//
// NiceMock, NaggyMock, and StrictMock "inherit" the constructors of
// their respective base class, with up-to $n arguments.  Therefore
// you can write NiceMock<MockFoo>(5, "a") to construct a nice mock
// where MockFoo has a constructor that accepts (int, const char*),
// for example.
//
// A known limitation is
 that NiceMock<MockFoo>, NaggyMock<MockFoo>,
// and StrictMock<MockFoo> only works for mock methods defined using
// the MOCK_METHOD* family of macros DIRECTLY in the MockFoo class.
// If a mock method is defined in a base class of MockFoo, the "nice"
// or "strict" modifier may not affect it, depending on the compiler.
// In particular, nesting NiceMock, NaggyMock, and StrictMock is NOT
// supported.
//
// Another known limitation is that the constructors of the base mock
// cannot have arguments passed by non-const reference, which are
// banned by the Google C++ style guide anyway.

#ifndef GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_NICE_STRICT_H_
#define GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_NICE_STRICT_H_

#include "gmock/gmock-spec-builders.h"
#include "gmock/internal/gmock-port.h"

namespace testing {

$range kind 0..2
$for kind [[

$var clazz=[[$if kind==0 [[NiceMock]]
          $elif kind==1 [[NaggyMock]]
          $else [[StrictMock]]]]

$var method=[[$if kind==0 [[AllowUninterestingCalls]]
           $elif kind==1 [[WarnUninterestingCalls]]
          $else [[FailUninterestingCalls]]]]

template <class MockClass>
class $clazz : public MockClass {
public:
 // We don't factor out the constructor body to a common method, as
 // we have to avoid a possible clash with members of MockClass.
 $clazz() {
```

```
  ::testing::Mock::$method(
      internal::ImplicitCast_<MockClass*>(this));
 }

 // C++ doesn't (yet) allow inheritance of constructors, so we have
 // to define it for each arity.
 template <typename A1>
 explicit $clazz(const A1& a1) : MockClass(a1) {
  ::testing::Mock::$method(
      internal::ImplicitCast_<MockClass*>(this));
 }

$range i 2..n
$for i [[
$range j 1..i
 template <$for j, [[typename A$j]]>
 $clazz($for j, [[const A$j& a$j]]) : MockClass($for j, [[a$j]]) {
  ::testing::Mock::$method(
      internal::ImplicitCast_<MockClass*>(this));
 }


]]
 virtual ~$clazz() {
  ::testing::Mock::UnregisterCallReaction(
      internal::ImplicitCast_<MockClass*>(this));
  }

private:
 GTEST_DISALLOW_COPY_AND_ASSIGN_($clazz);
};

]]

// The following specializations catch some (relatively more common)
// user errors of nesting nice and strict mocks.  They do NOT catch
// all possible errors.

// These specializations are declared but not defined, as NiceMock,
// NaggyMock, and StrictMock cannot be nested.

template <typename MockClass>
class NiceMock<NiceMock<MockClass> >;
template <typename MockClass>
class NiceMock<NaggyMock<MockClass> >;
template <typename MockClass>
class NiceMock<StrictMock<MockClass> >;
```

```
template <typename MockClass>
class NaggyMock<NiceMock<MockClass> >;
template <typename MockClass>
class NaggyMock<NaggyMock<MockClass> >;
template <typename MockClass>
class NaggyMock<StrictMock<MockClass> >;

template <typename MockClass>
class StrictMock<NiceMock<MockClass> >;
template <typename MockClass>
class StrictMock<NaggyMock<MockClass> >;
template <typename MockClass>
class StrictMock<StrictMock<MockClass>
 >;

}  // namespace testing

#endif  // GMOCK_INCLUDE_GMOCK_GMOCK_GENERATED_NICE_STRICT_H_
```

Found in path(s):
* /vendors-gtest-1-8-0-234-windows-zip/include/gmock/gmock-generated-nice-strict.h.pump
* /vendors-gtest-1-8-0-234-windows-zip/include/opensource/gmock-1.7.0/include/gmock/gmock-generated-nice-strict.h.pump

1.15 sipcc 12.8.0

1.15.1 Available under license :

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.

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.

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 HOLDER 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.

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 [2007] Neal Norwitz
Portions Copyright [2007] Google 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.

This file contains a list of people who've made non-trivial
contribution to the Google C++ Mocking Framework project. People
who commit code to the project are encouraged to add their names
here. Please keep the list sorted by first names.

Benoit Sigoure <tsuna@google.com>
Bogdan Piloca <boo@google.com>
Chandler Carruth <chandlerc@google.com>
Dave MacLachlan <dmaclach@gmail.com>
David Anderson <danderson@google.com>
Dean Sturtevant
Gene Volovich <gv@cite.com>
Hal Burch <gmock@hburch.com>
Jeffrey Yasskin <jyasskin@google.com>
Jim Keller <jimkeller@google.com>
Joe Walnes <joe@truemesh.com>
Jon Wray <jwray@google.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kostya Serebryany <kcc@google.com>
Lev Makhlis
Manuel Klimek <klimek@google.com>
Mario Tanev <radix@google.com>
Mark Paskin
Markus Heule <markus.heule@gmail.com>
Matthew Simmons <simmonmt@acm.org>
Mike Bland <mbland@google.com>
Neal Norwitz <nnorwitz@gmail.com>
Nermin Ozkiranartli <nermin@google.com>
Owen
 Carlsen <ocarlsen@google.com>
Paneendra Ba <paneendra@google.com>
Paul Menage <menage@google.com>
Piotr Kaminski <piotrk@google.com>
Russ Rufer <russ@pentad.com>
Sverre Sundsdal <sundsdal@gmail.com>
Takeshi Yoshino <tyoshino@google.com>
Vadim Berman <vadimb@google.com>
Vlad Losev <vladl@google.com>
Wolfgang Klier <wklier@google.com>
Zhanyong Wan <wan@google.com>

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.

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 HOLDER 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.
This file contains a list of people who've made non-trivial
contribution to the Google C++ Testing Framework project. People
who commit code to the project are encouraged to add their names
here. Please keep the list sorted by first names.

Ajay Joshi <jaj@google.com>
Balzs Dn <balazs.dan@gmail.com>
Bharat Mediratta <bharat@menalto.com>
Chandler Carruth <chandlerc@google.com>
Chris Prince <cprince@google.com>
Chris Taylor <taylorc@google.com>
Dan Egnor <egnor@google.com>
Eric Roman <eroman@chromium.org>
Hady Zalek <hady.zalek@gmail.com>
Jeffrey Yasskin <jyasskin@google.com>
Ji Sigursson <joi@google.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kenton Varda <kenton@google.com>
Manuel Klimek <klimek@google.com>
Markus Heule <markus.heule@gmail.com>
Mika Raento <mikie@iki.fi>
Mikls Fazekas <mfazekas@szemafor.com>
Pasi Valminen <pasi.valminen@gmail.com>
Patrick Hanna <phanna@google.com>
Patrick Riley <pfr@google.com>

Peter Kaminski <piotrk@google.com>
Preston
Jackson <preston.a.jackson@gmail.com>
Rainer Klaffenboeck <rainer.klaffenboeck@dynatrace.com>
Russ Cox <rsc@google.com>
Russ Rufer <russ@pentad.com>
Sean Mcafee <eefacm@gmail.com>
Sigurur sgeirsson <siggi@google.com>
Tracy Bialik <tracy@pentad.com>
Vadim Berman <vadimb@google.com>
Vlad Losev <vladl@google.com>
Zhanyong Wan <wan@google.com>

1.16 zlib 1.3.1

1.16.1 Available under license :

The MIT License (Expat)

Copyright (c) contributors

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.

1.17 nghttp2 1.64.0

1.17.1 Available under license :

nit Testing Framework
Copyright (c) 2013-2016 Evan Nemerson <evan@nemerson.com>

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) 2010- mruby developers

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.
The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa
Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

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.
LEGAL NOTICE INFORMATION

All the files in this distribution are covered under the MIT license
(see the file LICENSE) except some files mentioned below:

- src/readfloat.c: public domain by Yasuhiro Matsumoto (@mattn)
- src/fmt_fp.c: public domain by Dave Hylands (@dhylands)
- mrbgems/mruby-dir/src/Win/dirent.c: MIT-like license by Kevlin Henney

[src/readfloat.c]

strtod implementation.
author: Yasuhiro Matsumoto (@mattn)
license: public domain

The original code can be found in https://github.com/mattn/strtod

[src/fmt_fp.c]

The code in this function was inspired from Fred Bayer's pdouble.c.
Since pdouble.c was released as Public Domain, I'm releasing this
code as public domain as well.

Dave Hylands

The original code can be found in https://github.com/dhylands/format-float

[mrbgems/mruby-dir/src/Win/dirent.c] used only for Windows platform

Copyright Kevlin Henney, 1997, 2003, 2012. All rights reserved.

Permission to use, copy, modify, and distribute this

software and its
documentation for any purpose is hereby granted without fee, provided
that this copyright and permissions notice appear in all copies and
derivatives.

This software is supplied "as is" without express or implied warranty.

But that said, if there are any problems please get in touch.
set yrange [0:]
set terminal pngcairo font 'Sans, 8' lw 1 size 1400,1024
set xtics rotate by -45
set style histogram errorbars gap 2 lw 1
set style fill solid border -1
mruby-set

Copyright (c) yui-knk 2016

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.

[MIT license: http://www.opensource.org/licenses/mit-license.php]
If not otherwise noted, the extensions in this package are licensed
under the following license.

Copyright (c) 2010 by the contributors (see AUTHORS file).
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.

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.

1.18 openssl 1.1.1zb

1.18.1 Available under license :

LICENSE ISSUES
==============

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License

/* ==
* Copyright (c) 1998-2019 The OpenSSL Project. 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. All advertising materials mentioning features or use
of this
* software must display the following acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
* endorse or promote products derived from this software without
* prior written permission. For written permission, please contact
* openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
* nor may "OpenSSL" appear in their names without prior written
* permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
* acknowledgment:
* "This product includes software developed by the OpenSSL Project
* for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``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 THE OpenSSL PROJECT OR
* ITS 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.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* 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 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. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include
an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``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 AUTHOR 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.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this
 code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

GNU GENERAL PUBLIC LICENSE
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.

GNU GENERAL PUBLIC LICENSE
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

Appendix: 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) 19yy <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) 19yy 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.

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a
Package may be copied, such that the Copyright Holder maintains some
semblance of artistic control over the development of the package,
while giving the users of the package the right to use and distribute
the Package in a more-or-less customary fashion, plus the right to make
reasonable modifications.

Definitions:

 "Package" refers to the collection of files distributed by the

Copyright Holder, and derivatives of that collection of files
created through textual modification.

"Standard Version" refers to such a Package if it has not been
modified, or has been modified in accordance with the wishes
of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or
copyrights for the package.

"You" is you, if you're thinking about copying or distributing
this Package.

"Reasonable copying fee" is whatever you can justify on the
basis
of media cost, duplication charges, time of people involved,
and so on. (You will not be required to justify it to the
Copyright Holder, but only to the computing community at large
as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item
itself, though there may be fees involved in handling the item.
It also means that recipients of the item may redistribute it
under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the
Standard Version of this Package without restriction, provided that you
duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications
derived from the Public Domain or from the Copyright Holder. A Package
modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided
that you insert a prominent
notice in each changed file stating how and
when you changed that file, and provided that you do at least ONE of the
following:

a) place your modifications in the Public Domain or otherwise make them
Freely Available, such as by posting said modifications to Usenet or
an equivalent medium, or placing the modifications on a major archive
site such as uunet.uu.net, or by allowing the Copyright Holder to include
your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict

with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable
form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided
that you do not advertise this Package as a
product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines

(or comparably compiled subroutines in other
languages) supplied by you and linked into this Package in order to
emulate subroutines and variables of the language defined by this
Package shall not be considered part of this Package, but are the
equivalent of input as in Paragraph 6, provided these subroutines do
not change the language in any way that would cause it to fail the
regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always
permitted provided that the use of this Package is embedded; that is,
when no overt attempt is made to make this Package's interfaces visible
to the end user of the commercial distribution. Such use shall not be
construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote
products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT
LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

1.19 libxml2 2.14.2

1.19.1 Available under license :

Except where otherwise noted in the source code (e.g. the files dict.c and
list.c, which are covered by a similar licence but with different Copyright
notices) all the files are:

Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Copyright (C) The Libxml2 Contributors.

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 fur-
nished 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, FIT-
NESS

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.
The test cases in this directory are derived from the html5lib test
suite available from https://github.com/html5lib/html5lib-tests under
the following license:

Copyright (c) 2006-2013 James Graham, Geoffrey Sneddon, and
other contributors

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.

1.20 curl 8.14.1

1.20.1 Available under license :

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2025, Daniel Stenberg, <daniel@haxx.se>, and many
contributors, see the THANKS file.

All rights reserved.

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

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 OF THIRD PARTY RIGHTS. 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.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other
 dealings
in this Software without prior written authorization of the copyright holder.

1.21 sqlite 2025-07-17

1.21.1 Available under license :

The author disclaims copyright to this source code. In place of
a legal notice, here is a blessing:

  May you do good and not evil.
  May you find forgiveness for yourself and forgive others.
  May you share freely, never taking more than you give.
License Information
==========

SQLite Is Public Domain

The SQLite source code, including all of the files in the directories
listed in the bullets below are
[Public Domain](https://sqlite.org/copyright.html).
The authors have submitted written affidavits releasing their work to
the public for any use. Every byte of the public-domain code can be
traced back to the original authors. The files of this repository
that are public domain include the following:

 * All of the primary SQLite source code files found in the
   [src/ directory](https://sqlite.org/src/tree/src?type=tree&expand)
 * All of the test cases and testing code in the
   [test/ directory](https://sqlite.org/src/tree/test?type=tree&expand)
 * All of the SQLite extension source code and test cases in the
   [ext/ directory](https://sqlite.org/src/tree/ext?type=tree&expand)
 * All code that ends up in the "sqlite3.c" and "sqlite3.h" build products
   that actually implement
 the SQLite RDBMS.
 * All of the code used to compile the

[command-line interface](https://sqlite.org/cli.html)
* All of the code used to build various utility programs such as "sqldiff", "sqlite3_rsync", and "sqlite3_analyzer".

The public domain source files usually contain a header comment similar to the following to make it clear that the software is public domain.

> ~~~
The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:

* May you do good and not evil.
* May you find forgiveness for yourself and forgive others.
* May you share freely, never taking more than you give.
~~~

Almost every file you find in this source repository will be public domain. But there are a small number of exceptions:

Non-Public-Domain Code Included With This Source Repository AS A Convenience
---

This repository contains a (relatively) small amount of non-public-domain code
used to help implement the configuration and build logic. In other words, there are some non-public-domain files used to implement:

> ~~~
./configure && make
~~~

In all cases, the non-public-domain files included with this repository have generous BSD-style licenses. So anyone is free to use any of the code in this source repository for any purpose, though attribution may be required to reuse or republish the configure and build scripts. None of the non-public-domain code ever actually reaches the build products, such as "sqlite3.c", however, so no attribution is required to use SQLite itself. The non-public-domain code consists of scripts used to help compile SQLite. The non-public-domain code is technically not part of SQLite. The non-public-domain code is included in this repository as a convenience to developers, so that those who want to build SQLite do not need to go download a bunch of third-party build scripts in order to compile SQLite.

Non-public-domain

code included in this respository includes:

* The ["autosetup"](http://msteveb.github.io/autosetup/) configuration
  system that is contained (mostly) the autosetup/ directory, but also
  includes the "./configure" script at the top-level of this archive.
  Autosetup has a separate BSD-style license. See the
  [autosetup/LICENSE](http://msteveb.github.io/autosetup/license/)
  for details.

* There are BSD-style licenses on some of the configuration
  software found in the legacy autoconf/ directory and its
  subdirectories.

The following unix shell command is can be run from the top-level
of this source repository in order to remove all non-public-domain
code:

> ~~~
rm -rf configure autosetup autoconf
~~~

If you unpack this source repository and then run the command above, what
is left will be 100% public domain.
Unless explicitly stated, all files which form part of autosetup
are released under the following license:

---
autosetup - A build environment "autoconfigurator"

Copyright (c) 2010-2011, WorkWare Systems <http://workware.net.au/>

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.

THIS SOFTWARE IS PROVIDED BY THE WORKWARE SYSTEMS ``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 WORKWARE
SYSTEMS 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.

The views and conclusions contained in the software and documentation
are those of the authors and should not be interpreted as representing
official policies, either expressed or implied, of WorkWare Systems.

1.22 sqlite 2025-06-28

1.22.1 Available under license :

The author disclaims copyright to this source code. In place of
a legal notice, here is a blessing:

  May you do good and not evil.
  May you find forgiveness for yourself and forgive others.
  May you share freely, never taking more than you give.
License Information
==========

SQLite Is Public Domain

The SQLite source code, including all of the files in the directories
listed in the bullets below are
[Public Domain](https://sqlite.org/copyright.html).
The authors have submitted written affidavits releasing their work to
the public for any use. Every byte of the public-domain code can be
traced back to the original authors. The files of this repository
that are public domain include the following:

 * All of the primary SQLite source code files found in the
    [src/ directory](https://sqlite.org/src/tree/src?type=tree&expand)
 * All of the test cases and testing code in the
    [test/ directory](https://sqlite.org/src/tree/test?type=tree&expand)
 * All of the SQLite extension source code and test cases in the
    [ext/ directory](https://sqlite.org/src/tree/ext?type=tree&expand)
 * All code that ends up in the "sqlite3.c" and "sqlite3.h" build products
    that actually implement
 the SQLite RDBMS.
 * All of the code used to compile the
    [command-line interface](https://sqlite.org/cli.html)
 * All of the code used to build various utility programs such as

"sqldiff", "sqlite3_rsync", and "sqlite3_analyzer".

The public domain source files usually contain a header comment
similar to the following to make it clear that the software is
public domain.

> ~~~
The author disclaims copyright to this source code. In place of
a legal notice, here is a blessing:

 * May you do good and not evil.
 * May you find forgiveness for yourself and forgive others.
 * May you share freely, never taking more than you give.
~~~

Almost every file you find in this source repository will be
public domain. But there are a small number of exceptions:

Non-Public-Domain Code Included With This Source Repository AS A Convenience
---

This repository contains a (relatively) small amount of non-public-domain
code
 used to help implement the configuration and build logic. In other
words, there are some non-public-domain files used to implement:

> ~~~
./configure && make
~~~

In all cases, the non-public-domain files included with this
repository have generous BSD-style licenses. So anyone is free to
use any of the code in this source repository for any purpose, though
attribution may be required to reuse or republish the configure and
build scripts. None of the non-public-domain code ever actually reaches
the build products, such as "sqlite3.c", however, so no attribution is
required to use SQLite itself. The non-public-domain code consists of
scripts used to help compile SQLite. The non-public-domain code is
technically not part of SQLite. The non-public-domain code is
included in this repository as a convenience to developers, so that those
who want to build SQLite do not need to go download a bunch of
third-party build scripts in order to compile SQLite.

Non-public-domain
 code included in this respository includes:

 * The ["autosetup"](http://msteveb.github.io/autosetup/) configuration

system that is contained (mostly) the autosetup/ directory, but also
includes the "./configure" script at the top-level of this archive.
Autosetup has a separate BSD-style license. See the
[autosetup/LICENSE](http://msteveb.github.io/autosetup/license/)
for details.

* There are BSD-style licenses on some of the configuration
software found in the legacy autoconf/ directory and its
subdirectories.

The following unix shell command is can be run from the top-level
of this source repository in order to remove all non-public-domain
code:

> ~~~
rm -rf configure autosetup autoconf
~~~

If you unpack this source repository and then run the command above, what
is left will be 100% public domain.
Unless explicitly stated, all files which form part of autosetup
are released under the following license:

---
autosetup - A build environment "autoconfigurator"

Copyright (c) 2010-2011, WorkWare Systems <http://workware.net.au/>

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.

THIS SOFTWARE IS PROVIDED BY THE WORKWARE SYSTEMS ``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 WORKWARE
SYSTEMS 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.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of WorkWare Systems.

1.23 sentry 8.52.1

1.23.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Sentry

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.

1.24 fmdb 2.7.8

1.24.1 Available under license :

MIT License

Copyright (c) 2017-2021 Thomas Zoechling (https://www.peakstep.com)

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.

1.25 zipfoundation 0.9.16

1.25.1 Available under license :

MIT License

Copyright (c) 2017-2021 Thomas Zoechling (https://www.peakstep.com)

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.

1.26 fmt 14.29

1.26.1 Available under license :

A. HISTORY OF THE SOFTWARE
==========

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see http://www.zope.com). In 2001, the Python Software Foundation (PSF, see http://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of
the PSF.

All Python releases are Open Source (see http://www.opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release	Derived from	Year	Owner	GPL-compatible? (1)
0.9.0 thru 1.2		1991-1995	CWI	yes
1.3 thru 1.5.2	1.2	1995-1999	CNRI	yes
1.6	1.5.2	2000	CNRI	no
2.0	1.6	2000	BeOpen.com	no
1.6.1	1.6	2001	CNRI	yes (2)
2.1	2.0+1.6.1	2001	PSF	no
2.0.1	2.0+1.6.1	2001	PSF	yes
2.1.1	2.1+2.0.1	2001	PSF	yes
2.2	2.1.1	2001	PSF	yes
2.1.2	2.1.1	2002	PSF	yes
2.1.3	2.1.2	2002	PSF	yes
2.2.1	2.2	2002	PSF	yes
2.2.2	2.2.1	2002	PSF	yes
2.2.3	2.2.2	2003	PSF	yes
2.3	2.2.2	2002-2003	PSF	yes

2.3.1	2.3	2002-2003	PSF	yes
2.3.2	2.3.1	2002-2003	PSF	yes
2.3.3	2.3.2	2002-2003	PSF	yes
2.3.4	2.3.3	2004	PSF	yes
2.3.5	2.3.4	2005	PSF	yes
2.4	2.3	2004	PSF	yes
2.4.1	2.4	2005	PSF	yes
2.4.2	2.4.1	2005	PSF	yes
2.4.3	2.4.2	2006	PSF	yes
2.4.4	2.4.3	2006	PSF	yes
2.5	2.4	2006	PSF	yes
2.5.1	2.5	2007	PSF	yes
2.5.2	2.5.1	2008	PSF	yes
2.5.3	2.5.2	2008	PSF	yes
2.6	2.5	2008	PSF	yes
2.6.1	2.6	2008	PSF	yes
2.6.2	2.6.1	2009	PSF	yes
2.6.3	2.6.2	2009	PSF	yes
2.6.4	2.6.3	2009	PSF	yes
2.6.5	2.6.4	2010	PSF	yes
3.0	2.6	2008	PSF	yes
3.0.1	3.0	2009	PSF	yes
3.1	3.0.1	2009	PSF	yes
3.1.1	3.1	2009	PSF	yes
3.1.2	3.1.1	2010	PSF	yes
3.1.3	3.1.2	2010	PSF	yes
3.1.4	3.1.3	2011	PSF	yes
3.2	3.1	2011	PSF	yes
3.2.1	3.2	2011	PSF	yes
3.2.2	3.2.1	2011	PSF	yes
3.2.3	3.2.2	2012	PSF	yes
3.3.0	3.2	2012	PSF	yes

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's

direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON
==========

PYTHON
SOFTWARE FOUNDATION LICENSE VERSION 2
---

1. This LICENSE AGREEMENT is between the Python Software Foundation
("PSF"), and the Individual or Organization ("Licensee") accessing and
otherwise using this software ("Python") in source or binary form and
its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby
grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce,
analyze, test, perform and/or display publicly, prepare derivative works,
distribute, and otherwise use Python alone or in any derivative version,
provided, however, that PSF's License Agreement and PSF's notice of copyright,
i.e., "Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
2011, 2012 Python Software Foundation; All Rights Reserved" are retained in Python
alone or in any derivative version prepared by Licensee.

3. In the event Licensee
prepares a derivative work that is based on
or incorporates Python or any part thereof, and wants to make
the derivative work available to others as provided herein, then
Licensee hereby agrees to include in any such work a brief summary of
the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS"
basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON
FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

7.

Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee agrees to be bound by the terms and conditions of this License Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen
hereby grants Licensee a non-exclusive,
royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED
OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all

respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

1.
This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following
text (omitting the
quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the Internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS"
basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR
ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON
1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

7. This License Agreement shall be governed by the federal
intellectual property law of the United States, including without
limitation the federal copyright law, and, to the extent such
U.S. federal law does not apply, by the law of the Commonwealth of
Virginia, excluding Virginia's conflict of law provisions.
Notwithstanding the foregoing, with regard to derivative works based
on Python 1.6.1 that incorporate non-separable material that was
previously distributed under the GNU General Public License (GPL),
the
law of the Commonwealth of Virginia shall govern this License
Agreement only as to issues arising under or with respect to
Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this
License Agreement shall be deemed to create any relationship of
agency, partnership, or joint venture between CNRI and Licensee. This
License Agreement does not grant permission to use CNRI trademarks or
trade name in a trademark sense to endorse or promote products or
services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying,
installing or otherwise using Python 1.6.1, Licensee agrees to be
bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Copyright (c) 1991 - 1995, Stichting Mathematisch Centrum Amsterdam,
The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software
and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name of Stichting Mathematisch
Centrum or CWI not be used in advertising or publicity pertaining to
distribution of the software without specific, written prior
permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Copyright (c) 2012 - present, Victor Zverovich and {fmt} contributors

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.

--- Optional exception to the license ---

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into a machine-executable object form of such
source code, you may redistribute such embedded portions in such object form
without including the above copyright and permission notices.

partnership relationship between Cisco and any other company. (1110R)