Open Source Used In Network Insights Resource Analysis (NIR) 2.0.0

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: 78EE117C99-201043284

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

In your requests please include the following reference number 78EE117C99-201043284

Contents

1.1 buger/jsonparser master

1.1.1 Available under license :

MIT License

Copyright (c) 2016 Leonid Bugaev

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.

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 busybox 1.21.1 :7142499350fd020d6d0499a7ff65fc32

1.2.1 Available under license :

--- A note on GPL versions

BusyBox is distributed under version 2 of the General Public License (included in its entirety, below). Version 2 is the only version of this license which this version of BusyBox (or modified versions derived from this one) may be distributed under.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, 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.

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

How to Apply These Terms to Your New Programs

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

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

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

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

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

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
```

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

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year  name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
```

This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.

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

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

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

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

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
bzip2 applet in busybox is based on lightly-modified source
of bzip2 version 1.0.4. bzip2 source is distributed
under the following conditions (copied verbatim from LICENSE file)
===

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2006 Julian R Seward. 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. 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.

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

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

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "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 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.

Julian Seward, Cambridge, UK.
jseward@bzip.org
bzip2/libbzip2 version 1.0.4 of 20 December 2006

1.3 classnames 2.2.3

1.3.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Jed Watson

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.

1.4 classnames 2.2.3

1.4.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Jed Watson

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 (MIT)

Copyright (c) 2016 Jed Watson

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.

1.5 com.github.spullara.mustache.java/compiler 0.9.4

1.5.1 Available under license :

Copyright 2010 RightTime, Inc.

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

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

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

1.6 com.google.code.gson/gsob 2.7

1.6.1 Available under license :

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

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

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

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

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

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

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

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Google Gson

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 2008-2011 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.

1.7 com.google.protobuf/protobuf-java 3.7.1

1.7.1 Available under license :

Copyright 2008 Google Inc. All rights reserved.

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

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.
/// Json-cpp amalgated header (http://jsoncpp.sourceforge.net/).
/// It is intended to be used with #include "json/json.h"

// //
// Beginning of content of file: 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.

*/

// //////////////////////////////////////////////////////////////////////
// End of content of file: LICENSE
// //////////////////////////////////////////////////////////////////////





#ifndef JSON_AMALGATED_H_INCLUDED
# define JSON_AMALGATED_H_INCLUDED
/// If defined, indicates that the source file is amalgated
/// to prevent private header inclusion.
#define JSON_IS_AMALGAMATION

// //////////////////////////////////////////////////////////////////////
// Beginning of content of file: include/json/version.h
// //////////////////////////////////////////////////////////////////////

// DO NOT EDIT. This file (and "version") is generated by CMake.
// Run CMake configure step to update it.
#ifndef JSON_VERSION_H_INCLUDED
# define JSON_VERSION_H_INCLUDED

# define JSONCPP_VERSION_STRING "1.6.5"
# define JSONCPP_VERSION_MAJOR 1
# define JSONCPP_VERSION_MINOR 6
# define JSONCPP_VERSION_PATCH 5
# define JSONCPP_VERSION_QUALIFIER
# define JSONCPP_VERSION_HEXA ((JSONCPP_VERSION_MAJOR << 24) | (JSONCPP_VERSION_MINOR
<< 16) | (JSONCPP_VERSION_PATCH << 8))

#endif // JSON_VERSION_H_INCLUDED

// //////////////////////////////////////////////////////////////////////
// End of content of file: include/json/version.h
// //////////////////////////////////////////////////////////////////////
// //////////////////////////////////////////////////////////////////////
// Beginning of content of file: include/json/config.h
// //////////////////////////////////////////////////////////////////////

// Copyright 2007-2010 Baptiste Lepilleur
// Distributed under MIT license, or public domain if desired and
// recognized in your jurisdiction.
// See file LICENSE for detail or copy at http://jsoncpp.sourceforge.net/LICENSE

#ifndef JSON_CONFIG_H_INCLUDED
#define JSON_CONFIG_H_INCLUDED

/// If defined, indicates that json library is embedded in CppTL library.
//# define JSON_IN_CPPTL 1

/// If defined, indicates that json may leverage CppTL library
//#  define JSON_USE_CPPTL 1
/// If defined, indicates that cpptl vector based map should be used instead of
/// std::map
/// as Value container.
//#  define JSON_USE_CPPTL_SMALLMAP 1

// If non-zero, the library uses exceptions to report bad input instead of C
// assertion macros. The default is to use exceptions.
#ifndef JSON_USE_EXCEPTION
#define JSON_USE_EXCEPTION 1
#endif

/// If defined, indicates that the source file is amalgated
/// to prevent private header inclusion.
/// Remarks: it is automatically defined in the generated amalgated header.
// #define JSON_IS_AMALGAMATION

#ifdef JSON_IN_CPPTL
#include <cpptl/config.h>
#ifndef JSON_USE_CPPTL
#define JSON_USE_CPPTL 1
#endif
#endif

#ifdef JSON_IN_CPPTL
#define JSON_API CPPTL_API
#elif defined(JSON_DLL_BUILD)
#if defined(_MSC_VER)
#define JSON_API __declspec(dllexport)
#define JSONCPP_DISABLE_DLL_INTERFACE_WARNING
#endif // if defined(_MSC_VER)
```

```
#elif defined(JSON_DLL)
#if defined(_MSC_VER)
#define JSON_API __declspec(dllimport)
#define JSONCPP_DISABLE_DLL_INTERFACE_WARNING
#endif // if defined(_MSC_VER)
#endif // ifdef JSON_IN_CPPTL
#if !defined(JSON_API)
#define JSON_API
#endif

// If JSON_NO_INT64 is defined, then Json only support C++ "int" type for
// integer
// Storages, and 64 bits integer support is disabled.
// #define JSON_NO_INT64 1

#if defined(_MSC_VER) // MSVC
#  if _MSC_VER <= 1200 // MSVC 6
   // Microsoft Visual Studio 6 only support conversion from __int64 to double
   // (no conversion from unsigned __int64).
#    define JSON_USE_INT64_DOUBLE_CONVERSION 1
   // Disable warning 4786 for VS6 caused by STL (identifier was truncated to '255'
   // characters in the debug information)
   // All projects I've ever seen with VS6 were using this globally (not bothering
   // with pragma push/pop).
#    pragma warning(disable : 4786)
#  endif // MSVC 6

#  if _MSC_VER >= 1500 // MSVC 2008
   /// Indicates that the following function is deprecated.
#    define JSONCPP_DEPRECATED(message) __declspec(deprecated(message))
#  endif

#endif // defined(_MSC_VER)


#ifndef JSON_HAS_RVALUE_REFERENCES

#if defined(_MSC_VER) && _MSC_VER >= 1600 // MSVC >= 2010
#define JSON_HAS_RVALUE_REFERENCES 1
#endif // MSVC >= 2010

#ifdef __clang__
#if __has_feature(cxx_rvalue_references)
#define JSON_HAS_RVALUE_REFERENCES 1
#endif  // has_feature

#elif defined __GNUC__ // not clang (gcc comes later since clang emulates gcc)
#if defined(__GXX_EXPERIMENTAL_CXX0X__) || (__cplusplus >= 201103L)
```

```
#define JSON_HAS_RVALUE_REFERENCES 1
#endif  // GXX_EXPERIMENTAL

#endif // __clang__ || __GNUC__

#endif // not defined JSON_HAS_RVALUE_REFERENCES

#ifndef JSON_HAS_RVALUE_REFERENCES
#define JSON_HAS_RVALUE_REFERENCES 0
#endif

#ifdef __clang__
#elif defined __GNUC__ // not clang (gcc comes later since clang emulates gcc)
#  if (__GNUC__ > 4 || (__GNUC__ == 4 && __GNUC_MINOR__ >= 5))
#    define JSONCPP_DEPRECATED(message)  __attribute__ ((deprecated(message)))
#  elif (__GNUC__ > 3 || (__GNUC__ == 3 && __GNUC_MINOR__ >= 1))
#    define JSONCPP_DEPRECATED(message)  __attribute__((__deprecated__))
#  endif  // GNUC version
#endif // __clang__ || __GNUC__

#if !defined(JSONCPP_DEPRECATED)
#define JSONCPP_DEPRECATED(message)
#endif // if !defined(JSONCPP_DEPRECATED)

namespace Json {
typedef int Int;
typedef unsigned int UInt;
#if defined(JSON_NO_INT64)
typedef int LargestInt;
typedef unsigned int LargestUInt;
#undef JSON_HAS_INT64
#else                 // if defined(JSON_NO_INT64)
// For Microsoft Visual use specific types as long long is not supported
#if defined(_MSC_VER) // Microsoft Visual Studio
typedef __int64 Int64;
typedef unsigned __int64 UInt64;
#else                 // if defined(_MSC_VER) // Other platforms, use long long
typedef long long int Int64;
typedef unsigned long long int UInt64;
#endif // if defined(_MSC_VER)
typedef Int64 LargestInt;
typedef UInt64 LargestUInt;
#define JSON_HAS_INT64
#endif // if defined(JSON_NO_INT64)
} // end namespace Json

#endif // JSON_CONFIG_H_INCLUDED
```

```
// //////////////////////////////////////////////////////////////////////
// End of content of file: include/json/config.h
// //////////////////////////////////////////////////////////////////////






// //////////////////////////////////////////////////////////////////////
// Beginning of content of file: include/json/forwards.h
// //////////////////////////////////////////////////////////////////////

// Copyright 2007-2010 Baptiste Lepilleur
// Distributed under MIT license, or public domain if desired and
// recognized in your jurisdiction.
// See file LICENSE for detail or copy at http://jsoncpp.sourceforge.net/LICENSE

#ifndef JSON_FORWARDS_H_INCLUDED
#define JSON_FORWARDS_H_INCLUDED

#if !defined(JSON_IS_AMALGAMATION)
#include "config.h"
#endif // if !defined(JSON_IS_AMALGAMATION)

namespace Json {

// writer.h
class FastWriter;
class StyledWriter;

// reader.h
class Reader;

// features.h
class Features;

// value.h
typedef unsigned int ArrayIndex;
class StaticString;
class Path;
class PathArgument;
class Value;
class ValueIteratorBase;
class ValueIterator;
class ValueConstIterator;

} // namespace Json
```

```
#endif // JSON_FORWARDS_H_INCLUDED

// //////////////////////////////////////////////////////////////////////
// End of content of file: include/json/forwards.h
// //////////////////////////////////////////////////////////////////////






// //////////////////////////////////////////////////////////////////////
// Beginning of content of file: include/json/features.h
// //////////////////////////////////////////////////////////////////////

// Copyright 2007-2010 Baptiste Lepilleur
// Distributed under MIT license, or public domain if desired and
// recognized in your jurisdiction.
// See file LICENSE for detail or copy at http://jsoncpp.sourceforge.net/LICENSE

#ifndef CPPTL_JSON_FEATURES_H_INCLUDED
#define CPPTL_JSON_FEATURES_H_INCLUDED

#if !defined(JSON_IS_AMALGAMATION)
#include "forwards.h"
#endif // if !defined(JSON_IS_AMALGAMATION)

namespace Json {

/** \brief Configuration passed to reader and writer.
* This configuration object can be used to force the Reader or Writer
* to behave in a standard conforming way.
*/
class JSON_API Features {
public:
 /** \brief A configuration that allows all features and assumes all strings
  * are UTF-8.
  * - C & C++ comments are allowed
  * - Root object can be any JSON value
  * - Assumes Value strings are encoded in UTF-8
  */
 static Features all();

 /** \brief A configuration that is strictly compatible with the JSON
  * specification.
  * - Comments are forbidden.
  * - Root object must be either an array or an object value.
  * - Assumes Value strings are encoded in UTF-8
  */
 static Features strictMode();

 /** \brief Initialize the configuration like JsonConfig::allFeatures;
  */
 Features();

 /// \c true if comments are allowed. Default: \c true.
 bool allowComments_;

 /// \c true if root must be either an array or an object value. Default: \c
 /// false.
 bool strictRoot_;

 /// \c true if dropped null placeholders are allowed. Default: \c false.
 bool allowDroppedNullPlaceholders_;

 /// \c true if numeric object key are allowed. Default: \c false.
 bool allowNumericKeys_;
};

} // namespace Json

#endif // CPPTL_JSON_FEATURES_H_INCLUDED

// //////////////////////////////////////////////////////////////////////
// End of content of file: include/json/features.h
// //////////////////////////////////////////////////////////////////////






// //////////////////////////////////////////////////////////////////////
// Beginning of content of file: include/json/value.h
// //////////////////////////////////////////////////////////////////////

// Copyright 2007-2010 Baptiste Lepilleur
// Distributed under MIT license, or public domain if desired and
// recognized in your jurisdiction.
// See file LICENSE for detail or copy at http://jsoncpp.sourceforge.net/LICENSE

#ifndef CPPTL_JSON_H_INCLUDED
#define CPPTL_JSON_H_INCLUDED

#if !defined(JSON_IS_AMALGAMATION)
```

```
#include "forwards.h"
#endif // if !defined(JSON_IS_AMALGAMATION)
#include <string>
#include <vector>
#include <exception>

#ifndef JSON_USE_CPPTL_SMALLMAP
#include <map>
#else
#include <cpptl/smallmap.h>
#endif
#ifdef JSON_USE_CPPTL
#include <cpptl/forwards.h>
#endif

// Disable warning C4251: <data member>: <type> needs to have dll-interface to
// be used by...
#if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)
#pragma warning(push)
#pragma warning(disable : 4251)
#endif // if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)

/** \brief JSON (JavaScript Object Notation).
*/
namespace Json {

/** Base class for all exceptions we throw.
*
* We use nothing but these internally. Of course, STL can throw others.
*/
class JSON_API Exception : public std::exception {
public:
 Exception(std::string const& msg);
 ~Exception() throw() override;
 char const* what() const throw() override;
protected:
 std::string msg_;
};

/** Exceptions which the user cannot easily avoid.
*
* E.g. out-of-memory (when we use malloc), stack-overflow, malicious input
*
* \remark derived from Json::Exception
*/
class JSON_API RuntimeError : public Exception {
public:
 RuntimeError(std::string const& msg);
};

/** Exceptions thrown by JSON_ASSERT/JSON_FAIL macros.
*
* These are precondition-violations (user bugs) and internal errors (our bugs).
*
* \remark derived from Json::Exception
*/
class JSON_API LogicError : public Exception {
public:
 LogicError(std::string const& msg);
};

/// used internally
void throwRuntimeError(std::string const& msg);
/// used internally
void throwLogicError(std::string const& msg);

/** \brief Type of the value held by a Value object.
*/
enum ValueType {
 nullValue = 0, ///< 'null' value
 intValue,      ///< signed integer value
 uintValue,     ///< unsigned integer value
 realValue,     ///< double value
 stringValue,   ///< UTF-8 string value
 booleanValue,  ///< bool value
 arrayValue,    ///< array value (ordered list)
 objectValue    ///< object value (collection of name/value pairs).
};

enum CommentPlacement {
 commentBefore = 0,      ///< a comment placed on the line before a value
 commentAfterOnSameLine, ///< a comment just after a value on the same line
 commentAfter, ///< a comment on the line after a value (only make sense for
 /// root value)
 numberOfCommentPlacement
};

//# ifdef JSON_USE_CPPTL
//   typedef CppTL::AnyEnumerator<const char *> EnumMemberNames;
//   typedef CppTL::AnyEnumerator<const Value &> EnumValues;
//# endif

/** \brief Lightweight wrapper to tag static string.
*
* Value constructor and objectValue member assignement takes advantage of the
* StaticString and avoid the cost of string duplication when storing the
* string or the member name.
*
* Example of usage:
* \code
* Json::Value aValue( StaticString("some text") );
* Json::Value object;
* static const StaticString code("code");
* object[code] = 1234;
* \endcode
*/
class JSON_API StaticString {
public:
 explicit StaticString(const char* czstring) : c_str_(czstring) {}

 operator const char*() const { return c_str_; }

 const char* c_str() const { return c_str_; }

private:
 const char* c_str_;
};

/** \brief Represents a <a HREF="http://www.json.org">JSON</a> value.
*
* This class is a discriminated union wrapper that can represents a:
* - signed integer [range: Value::minInt - Value::maxInt]
* - unsigned integer (range: 0 - Value::maxUInt)
* - double
* - UTF-8 string
* - boolean
* - 'null'
* - an ordered list of Value
* - collection of name/value pairs (javascript object)
*
* The type of the held value is represented by a #ValueType and
* can be obtained using type().
*
* Values of an #objectValue or #arrayValue can be accessed using operator[]()
* methods.
* Non-const methods will automatically create the a #nullValue element
* if it does not exist.
* The sequence of an #arrayValue will be automatically resized and initialized
* with #nullValue. resize() can be used to enlarge or truncate an #arrayValue.
*
* The get() methods can be used to obtain default value in the case the
* required element does not exist.
*
* It is possible to iterate over the list of a #objectValue values using
* the getMemberNames() method.
*
* \note #Value string-length fit in size_t, but keys must be < 2^30.
* (The reason is an implementation detail.) A #CharReader will raise an
* exception if a bound is exceeded to avoid security holes in your app,
* but the Value API does *not* check bounds. That is the responsibility
* of the caller.
*/
class JSON_API Value {
 friend class ValueIteratorBase;
public:
 typedef std::vector<std::string> Members;
 typedef ValueIterator iterator;
 typedef ValueConstIterator const_iterator;
 typedef Json::UInt UInt;
 typedef Json::Int Int;
#if defined(JSON_HAS_INT64)
 typedef Json::UInt64 UInt64;
 typedef Json::Int64 Int64;
#endif // defined(JSON_HAS_INT64)
 typedef Json::LargestInt LargestInt;
 typedef Json::LargestUInt LargestUInt;
 typedef Json::ArrayIndex ArrayIndex;

 static const Value& null;  ///< We regret this reference to a global instance; prefer the simpler Value().
 static const Value& nullRef;  ///< just a kludge for binary-compatibility; same as null
 /// Minimum signed integer value that can be stored in a Json::Value.
 static const LargestInt minLargestInt;
 /// Maximum signed integer value that can be stored in a Json::Value.
 static const LargestInt maxLargestInt;
 /// Maximum unsigned integer value that can be stored in a Json::Value.
 static const LargestUInt maxLargestUInt;

 /// Minimum signed int value that can be stored in a Json::Value.
 static const Int minInt;
 /// Maximum signed int value that can be stored in a Json::Value.
 static const Int maxInt;
 /// Maximum unsigned int value that can be stored in a Json::Value.
 static const UInt maxUInt;

#if defined(JSON_HAS_INT64)
 /// Minimum signed 64 bits int value that can be stored in a Json::Value.
 static const Int64 minInt64;
 /// Maximum signed 64 bits int value that can be stored in a Json::Value.
 static const Int64 maxInt64;
 /// Maximum unsigned 64 bits int value that can be stored in a Json::Value.
 static const UInt64 maxUInt64;
#endif // defined(JSON_HAS_INT64)
```

```
private:
#ifndef JSONCPP_DOC_EXCLUDE_IMPLEMENTATION
 class CZString {
 public:
   enum DuplicationPolicy {
     noDuplication = 0,
     duplicate,
     duplicateOnCopy
   };
   CZString(ArrayIndex index);
   CZString(char const* str, unsigned length, DuplicationPolicy allocate);
   CZString(CZString const& other);
#if JSON_HAS_RVALUE_REFERENCES
   CZString(CZString&& other);
#endif
   ~CZString();
   CZString& operator=(CZString other);
   bool operator<(CZString const& other) const;
   bool operator==(CZString const& other) const;
   ArrayIndex index() const;
   //const char* c_str() const; ///< \deprecated
   char const* data() const;
   unsigned length() const;
   bool isStaticString() const;

 private:
   void swap(CZString& other);

   struct StringStorage {
     unsigned policy_: 2;
     unsigned length_: 30; // 1GB max
   };

   char const* cstr_;  // actually, a prefixed string, unless policy is noDup
   union {
     ArrayIndex index_;
     StringStorage storage_;
   };
 };

public:
#ifndef JSON_USE_CPPTL_SMALLMAP
 typedef std::map<CZString, Value> ObjectValues;
#else
 typedef CppTL::SmallMap<CZString, Value> ObjectValues;
#endif // ifndef JSON_USE_CPPTL_SMALLMAP
#endif // ifndef JSONCPP_DOC_EXCLUDE_IMPLEMENTATION
```

```
public:
 /** \brief Create a default Value of the given type.

   This is a very useful constructor.
   To create an empty array, pass arrayValue.
   To create an empty object, pass objectValue.
   Another Value can then be set to this one by assignment.
This is useful since clear() and resize() will not alter types.

   Examples:
\code
Json::Value null_value; // null
Json::Value arr_value(Json::arrayValue); // []
Json::Value obj_value(Json::objectValue); // {}
\endcode
 */
 Value(ValueType type = nullValue);
 Value(Int value);
 Value(UInt value);
#if defined(JSON_HAS_INT64)
 Value(Int64 value);
 Value(UInt64 value);
#endif // if defined(JSON_HAS_INT64)
 Value(double value);
 Value(const char* value); ///< Copy til first 0. (NULL causes to seg-fault.)
 Value(const char* begin, const char* end); ///< Copy all, incl zeroes.
 /** \brief Constructs a value from a static string.

  * Like other value string constructor but do not duplicate the string for
  * internal storage. The given string must remain alive after the call to this
  * constructor.
  * \note This works only for null-terminated strings. (We cannot change the
  *   size of this class, so we have nowhere to store the length,
  *   which might be computed later for various operations.)
  *
  * Example of usage:
  * \code
  * static StaticString foo("some text");
  * Json::Value aValue(foo);
  * \endcode
  */
 Value(const StaticString& value);
 Value(const std::string& value); ///< Copy data() til size(). Embedded zeroes too.
#ifdef JSON_USE_CPPTL
 Value(const CppTL::ConstString& value);
#endif
 Value(bool value);
 /// Deep copy.
 Value(const Value& other);
#if JSON_HAS_RVALUE_REFERENCES
 /// Move constructor
 Value(Value&& other);
#endif
 ~Value();

 /// Deep copy, then swap(other).
 /// \note Over-write existing comments. To preserve comments, use #swapPayload().
 Value& operator=(Value other);
 /// Swap everything.
 void swap(Value& other);
 /// Swap values but leave comments and source offsets in place.
 void swapPayload(Value& other);

 ValueType type() const;

 /// Compare payload only, not comments etc.
 bool operator<(const Value& other) const;
 bool operator<=(const Value& other) const;
 bool operator>=(const Value& other) const;
 bool operator>(const Value& other) const;
 bool operator==(const Value& other) const;
 bool operator!=(const Value& other) const;
 int compare(const Value& other) const;

 const char* asCString() const; ///< Embedded zeroes could cause you trouble!
 std::string asString() const; ///< Embedded zeroes are possible.
 /** Get raw char* of string-value.
  *  \return false if !string. (Seg-fault if str or end are NULL.)
  */
 bool getString(
     char const** begin, char const** end) const;
#ifdef JSON_USE_CPPTL
 CppTL::ConstString asConstString() const;
#endif
 Int asInt() const;
 UInt asUInt() const;
#if defined(JSON_HAS_INT64)
 Int64 asInt64() const;
 UInt64 asUInt64() const;
#endif // if defined(JSON_HAS_INT64)
 LargestInt asLargestInt() const;
 LargestUInt asLargestUInt() const;
 float asFloat() const;
 double asDouble() const;
 bool asBool() const;
 bool isNull() const;
 bool isBool() const;
 bool isInt() const;
 bool isInt64() const;
 bool isUInt() const;
 bool isUInt64() const;
 bool isIntegral() const;
 bool isDouble() const;
 bool isNumeric() const;
 bool isString() const;
 bool isArray() const;
 bool isObject() const;

 bool isConvertibleTo(ValueType other) const;

 /// Number of values in array or object
 ArrayIndex size() const;

 /// \brief Return true if empty array, empty object, or null;
 /// otherwise, false.
 bool empty() const;

 /// Return isNull()
 bool operator!() const;

 /// Remove all object members and array elements.
 /// \pre type() is arrayValue, objectValue, or nullValue
 /// \post type() is unchanged
 void clear();

 /// Resize the array to size elements.
 /// New elements are initialized to null.
 /// May only be called on nullValue or arrayValue.
 /// \pre type() is arrayValue or nullValue
 /// \post type() is arrayValue
 void resize(ArrayIndex size);

 /// Access an array element (zero based index ).
 /// If the array contains less than index element, then null value are
 /// inserted
 /// in the array so that its size is index+1.
 /// (You may need to say 'value[0u]' to get your compiler to distinguish
 ///  this from the operator[] which takes a string.)
 Value& operator[](ArrayIndex index);

 /// Access an array element (zero based index ).
 /// If the array contains less than index element, then null value are
 /// inserted
 /// in the array so that its size is index+1.
 /// (You may need to say 'value[0u]' to get your compiler to distinguish
 ///  this from the operator[] which takes a string.)
 Value& operator[](int index);

 /// Access an array element (zero based index )
 /// (You may need to say 'value[0u]' to get your compiler to distinguish
 ///  this from the operator[] which takes a string.)
 const Value& operator[](ArrayIndex index) const;

 /// Access an array element (zero based index )
 /// (You may need to say 'value[0u]' to get your compiler to distinguish
 ///  this from the operator[] which takes a string.)
 const Value& operator[](int index) const;

 /// If the array contains at least index+1 elements, returns the element
 /// value,
 /// otherwise returns defaultValue.
 Value get(ArrayIndex index, const Value& defaultValue) const;
 /// Return true if index < size().
 bool isValidIndex(ArrayIndex index) const;
 /// \brief Append value to array at the end.
 ///
 /// Equivalent to jsonvalue[jsonvalue.size()] = value;
 Value& append(const Value& value);

 /// Access an object value by name, create a null member if it does not exist.
 /// \note Because of our implementation, keys are limited to 2^30 -1 chars.
 ///  Exceeding that will cause an exception.
 Value& operator[](const char* key);
 /// Access an object value by name, returns null if there is no member with
 /// that name.
 const Value& operator[](const char* key) const;
 /// Access an object value by name, create a null member if it does not exist.
 /// \param key may contain embedded nulls.
 Value& operator[](const std::string& key);
 /// Access an object value by name, returns null if there is no member with
 /// that name.
 /// \param key may contain embedded nulls.
 const Value& operator[](const std::string& key) const;
 /** \brief Access an object value by name, create a null member if it does not
  exist.

  * If the object has no entry for that name, then the member name used to store
  * the new entry is not duplicated.
  * Example of use:
  * \code
```

```
  * Json::Value object;
  * static const StaticString code("code");
  * object[code] = 1234;
  * \endcode
  */
 Value& operator[](const StaticString& key);
#ifdef JSON_USE_CPPTL
 /// Access an object value by name, create a null member if it does not exist.
 Value& operator[](const CppTL::ConstString& key);
 /// Access an object value by name, returns null if there is no member with
 /// that name.
 const Value& operator[](const CppTL::ConstString& key) const;
#endif
 /// Return the member named key if it exist, defaultValue otherwise.
 /// \note deep copy
 Value get(const char* key, const Value& defaultValue) const;
 /// Return the member named key if it exist, defaultValue otherwise.
 /// \note deep copy
 /// \note key may contain embedded nulls.
 Value get(const char* begin, const char* end, const Value& defaultValue) const;
 /// Return the member named key if it exist, defaultValue otherwise.
 /// \note deep copy
 /// \param key may contain embedded nulls.
 Value get(const std::string& key, const Value& defaultValue) const;
#ifdef JSON_USE_CPPTL
 /// Return the member named key if it exist, defaultValue otherwise.
 /// \note deep copy
 Value get(const CppTL::ConstString& key, const Value& defaultValue) const;
#endif
 /// Most general and efficient version of isMember()const, get()const,
 /// and operator[]const
 /// \note As stated elsewhere, behavior is undefined if (end-begin) >= 2^30
 Value const* find(char const* begin, char const* end) const;
 /// Most general and efficient version of object-mutators.
 /// \note As stated elsewhere, behavior is undefined if (end-begin) >= 2^30
 /// \return non-zero, but JSON_ASSERT if this is neither object nor nullValue.
 Value const* demand(char const* begin, char const* end);
 /// \brief Remove and return the named member.
 ///
 /// Do nothing if it did not exist.
 /// \return the removed Value, or null.
 /// \pre type() is objectValue or nullValue
 /// \post type() is unchanged
 /// \deprecated
 Value removeMember(const char* key);
 /// Same as removeMember(const char*)
 /// \param key may contain embedded nulls.
 /// \deprecated
 Value removeMember(const std::string& key);
 /// Same as removeMember(const char* begin, const char* end, Value* removed),
 /// but 'key' is null-terminated.
 bool removeMember(const char* key, Value* removed);
 /** \brief Remove the named map member.

     Update 'removed' iff removed.
     \param key may contain embedded nulls.
     \return true iff removed (no exceptions)
 */
 bool removeMember(std::string const& key, Value* removed);
 /// Same as removeMember(std::string const& key, Value* removed)
 bool removeMember(const char* begin, const char* end, Value* removed);
 /** \brief Remove the indexed array element.

     O(n) expensive operations.
     Update 'removed' iff removed.
     \return true iff removed (no exceptions)
 */
 bool removeIndex(ArrayIndex i, Value* removed);

 /// Return true if the object has a member named key.
 /// \note 'key' must be null-terminated.
 bool isMember(const char* key) const;
 /// Return true if the object has a member named key.
 /// \param key may contain embedded nulls.
 bool isMember(const std::string& key) const;
 /// Same as isMember(std::string const& key)const
 bool isMember(const char* begin, const char* end) const;
#ifdef JSON_USE_CPPTL
 /// Return true if the object has a member named key.
 bool isMember(const CppTL::ConstString& key) const;
#endif

 /// \brief Return a list of the member names.
 ///
 /// If null, return an empty list.
 /// \pre type() is objectValue or nullValue
 /// \post if type() was nullValue, it remains nullValue
 Members getMemberNames() const;

 //# ifdef JSON_USE_CPPTL
 //      EnumMemberNames enumMemberNames() const;
 //      EnumValues enumValues() const;
 //# endif

 /// \deprecated Always pass len.
 JSONCPP_DEPRECATED("Use setComment(std::string const&) instead.")
 void setComment(const char* comment, CommentPlacement placement);
 /// Comments must be //... or /* ... */
 void setComment(const char* comment, size_t len, CommentPlacement placement);
 /// Comments must be //... or /* ... */
 void setComment(const std::string& comment, CommentPlacement placement);
 bool hasComment(CommentPlacement placement) const;
 /// Include delimiters and embedded newlines.
 std::string getComment(CommentPlacement placement) const;

 std::string toStyledString() const;

 const_iterator begin() const;
 const_iterator end() const;

 iterator begin();
 iterator end();

 // Accessors for the [start, limit) range of bytes within the JSON text from
 // which this value was parsed, if any.
 void setOffsetStart(size_t start);
 void setOffsetLimit(size_t limit);
 size_t getOffsetStart() const;
 size_t getOffsetLimit() const;

private:
 void initBasic(ValueType type, bool allocated = false);

 Value& resolveReference(const char* key);
 Value& resolveReference(const char* key, const char* end);

 struct CommentInfo {
   CommentInfo();
   ~CommentInfo();

   void setComment(const char* text, size_t len);

   char* comment_;
 };

 // struct MemberNamesTransform
 //{
 //   typedef const char *result_type;
 //   const char *operator()( const CZString &name ) const
 //   {
 //      return name.c_str();
 //   }
 //};
 union ValueHolder {
   LargestInt int_;
   LargestUInt uint_;
   double real_;
   bool bool_;
   char* string_;  // actually ptr to unsigned, followed by str, unless !allocated_
   ObjectValues* map_;
 } value_;
 ValueType type_ : 8;
 unsigned int allocated_ : 1; // Notes: if declared as bool, bitfield is useless.
                              // If not allocated_, string_ must be null-terminated.
 CommentInfo* comments_;

 // [start, limit) byte offsets in the source JSON text from which this Value
 // was extracted.
 size_t start_;
 size_t limit_;
};

/** \brief Experimental and untested: represents an element of the "path" to
* access a node.
*/
class JSON_API PathArgument {
public:
 friend class Path;

 PathArgument();
 PathArgument(ArrayIndex index);
 PathArgument(const char* key);
 PathArgument(const std::string& key);

private:
 enum Kind {
   kindNone = 0,
   kindIndex,
   kindKey
 };
 std::string key_;
 ArrayIndex index_;
 Kind kind_;
};

/** \brief Experimental and untested: represents a "path" to access a node.
*
* Syntax:
* - "." => root node
* - ".[n]" => elements at index 'n' of root node (an array value)
* - ".name" => member named 'name' of root node (an object value)
* - ".name1.name2.name3"
* - ".[0][1][2].name1[3]"
* - ".%" => member name is provided as parameter
* - ".[%]" => index is provied as parameter
*/
class JSON_API Path {
public:
 Path(const std::string& path,
      const PathArgument& a1 = PathArgument(),
      const PathArgument& a2 = PathArgument(),
      const PathArgument& a3 = PathArgument(),
      const PathArgument& a4 = PathArgument(),
      const PathArgument& a5 = PathArgument());

 const Value& resolve(const Value& root) const;
 Value resolve(const Value& root, const Value& defaultValue) const;
 /// Creates the "path" to access the specified node and returns a reference on
 /// the node.
 Value& make(Value& root) const;

private:
 typedef std::vector<const PathArgument*> InArgs;
 typedef std::vector<PathArgument> Args;

 void makePath(const std::string& path, const InArgs& in);
 void addPathInArg(const std::string& path,
                   const InArgs& in,
                   InArgs::const_iterator& itInArg,
                   PathArgument::Kind kind);
 void invalidPath(const std::string& path, int location);

 Args args_;
};

/** \brief base class for Value iterators.
*
*/
class JSON_API ValueIteratorBase {
public:
 typedef std::bidirectional_iterator_tag iterator_category;
 typedef unsigned int size_t;
 typedef int difference_type;
 typedef ValueIteratorBase SelfType;

 bool operator==(const SelfType& other) const { return isEqual(other); }

 bool operator!=(const SelfType& other) const { return !isEqual(other); }
 difference_type operator-(const SelfType& other) const {
   return other.computeDistance(*this);
 }

 /// Return either the index or the member name of the referenced value as a
 /// Value.
 Value key() const;

 /// Return the index of the referenced Value, or -1 if it is not an arrayValue.
 UInt index() const;

 /// Return the member name of the referenced Value, or "" if it is not an
 /// objectValue.
 /// \note Avoid `c_str()` on result, as embedded zeroes are possible.
 std::string name() const;

 /// Return the member name of the referenced Value. "" if it is not an
 /// objectValue.
 /// \deprecated This cannot be used for UTF-8 strings, since there can be embedded nulls.
 JSONCPP_DEPRECATED("Use `key = name();` instead.")
 char const* memberName() const;
 /// Return the member name of the referenced Value, or NULL if it is not an
 /// objectValue.
 /// \note Better version than memberName(). Allows embedded nulls.
 char const* memberName(char const** end) const;

protected:
 Value& deref() const;

 void increment();

 void decrement();

 difference_type computeDistance(const SelfType& other) const;

 bool isEqual(const SelfType& other) const;

 void copy(const SelfType& other);

private:
 Value::ObjectValues::iterator current_;
 // Indicates that iterator is for a null value.
 bool isNull_;

public:
 // For some reason, BORLAND needs these at the end, rather
 // than earlier. No idea why.
 ValueIteratorBase();
 explicit ValueIteratorBase(const Value::ObjectValues::iterator& current);
};

/** \brief const iterator for object and array value.
*
*/
class JSON_API ValueConstIterator : public ValueIteratorBase {
 friend class Value;

public:
 typedef const Value value_type;
 //typedef unsigned int size_t;
 //typedef int difference_type;
 typedef const Value& reference;
 typedef const Value* pointer;
 typedef ValueConstIterator SelfType;

 ValueConstIterator();
 ValueConstIterator(ValueIterator const& other);

private:
/*! \internal Use by Value to create an iterator.
*/
 explicit ValueConstIterator(const Value::ObjectValues::iterator& current);
public:
 SelfType& operator=(const ValueIteratorBase& other);

 SelfType operator++(int) {
   SelfType temp(*this);
   ++*this;
   return temp;
 }

 SelfType operator--(int) {
   SelfType temp(*this);
   --*this;
   return temp;
 }

 SelfType& operator--() {
   decrement();
   return *this;
 }

 SelfType& operator++() {
   increment();
   return *this;
 }
 reference operator*() const { return deref(); }

 pointer operator->() const { return &deref(); }
};

/** \brief Iterator for object and array value.
*/
class JSON_API ValueIterator : public ValueIteratorBase {
 friend class Value;

public:
 typedef Value value_type;
 typedef unsigned int size_t;
 typedef int difference_type;
 typedef Value& reference;
 typedef Value* pointer;
 typedef ValueIterator SelfType;

 ValueIterator();
 explicit ValueIterator(const ValueConstIterator& other);
 ValueIterator(const ValueIterator& other);

private:
/*! \internal Use by Value to create an iterator.
*/
 explicit ValueIterator(const Value::ObjectValues::iterator& current);
public:
 SelfType& operator=(const SelfType& other);

 SelfType operator++(int) {
   SelfType temp(*this);
   ++*this;
   return temp;
 }

 SelfType operator--(int) {
   SelfType temp(*this);
   --*this;
   return temp;
 }

 SelfType& operator--() {
   decrement();
   return *this;
 }

 SelfType& operator++() {
   increment();
   return *this;
 }

 reference operator*() const { return deref(); }

 pointer operator->() const { return &deref(); }
};

} // namespace Json


namespace std {
/// Specialize std::swap() for Json::Value.
template<>
inline void swap(Json::Value& a, Json::Value& b) { a.swap(b); }
}


#if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)
#pragma warning(pop)
#endif // if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)

#endif // CPPTL_JSON_H_INCLUDED

// //////////////////////////////////////////////////////////////////////
// End of content of file: include/json/value.h
// //////////////////////////////////////////////////////////////////////






// //////////////////////////////////////////////////////////////////////
// Beginning of content of file: include/json/reader.h
// //////////////////////////////////////////////////////////////////////

// Copyright 2007-2010 Baptiste Lepilleur
// Distributed under MIT license, or public domain if desired and
// recognized in your jurisdiction.
// See file LICENSE for detail or copy at http://jsoncpp.sourceforge.net/LICENSE

#ifndef CPPTL_JSON_READER_H_INCLUDED
#define CPPTL_JSON_READER_H_INCLUDED

#if !defined(JSON_IS_AMALGAMATION)
#include "features.h"
```

```
#include "value.h"
#endif // if !defined(JSON_IS_AMALGAMATION)
#include <deque>
#include <iosfwd>
#include <stack>
#include <string>
#include <istream>

// Disable warning C4251: <data member>: <type> needs to have dll-interface to
// be used by...
#if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)
#pragma warning(push)
#pragma warning(disable : 4251)
#endif // if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)

namespace Json {

/** \brief Unserialize a <a HREF="http://www.json.org">JSON</a> document into a
*Value.
*
* \deprecated Use CharReader and CharReaderBuilder.
*/
class JSON_API Reader {
public:
 typedef char Char;
 typedef const Char* Location;

 /** \brief An error tagged with where in the JSON text it was encountered.
  *
  * The offsets give the [start, limit) range of bytes within the text. Note
  * that this is bytes, not codepoints.
  *
  */
 struct StructuredError {
   size_t offset_start;
   size_t offset_limit;
   std::string message;
 };

 /** \brief Constructs a Reader allowing all features
  * for parsing.
  */
 Reader();

 /** \brief Constructs a Reader allowing the specified feature set
  * for parsing.
  */
 Reader(const Features& features);
```

```
/** \brief Read a Value from a <a HREF="http://www.json.org">JSON</a>
 * document.
 * \param document UTF-8 encoded string containing the document to read.
 * \param root [out] Contains the root value of the document if it was
 *             successfully parsed.
 * \param collectComments \c true to collect comment and allow writing them
 * back during
 *                        serialization, \c false to discard comments.
 *                        This parameter is ignored if
 * Features::allowComments_
 *                        is \c false.
 * \return \c true if the document was successfully parsed, \c false if an
 * error occurred.
 */
bool
parse(const std::string& document, Value& root, bool collectComments = true);

/** \brief Read a Value from a <a HREF="http://www.json.org">JSON</a>
 document.
 * \param beginDoc Pointer on the beginning of the UTF-8 encoded string of the
 document to read.
 * \param endDoc Pointer on the end of the UTF-8 encoded string of the
 document to read.
 *               Must be >= beginDoc.
 * \param root [out] Contains the root value of the document if it was
 *             successfully parsed.
 * \param collectComments \c true to collect comment and allow writing them
 back during
 *                        serialization, \c false to discard comments.
 *                        This parameter is ignored if
 Features::allowComments_
 *                        is \c false.
 * \return \c true if the document was successfully parsed, \c false if an
 error occurred.
 */
bool parse(const char* beginDoc,
           const char* endDoc,
           Value& root,
           bool collectComments = true);

/// \brief Parse from input stream.
/// \see Json::operator>>(std::istream&, Json::Value&).
bool parse(std::istream& is, Value& root, bool collectComments = true);

/** \brief Returns a user friendly string that list errors in the parsed
 * document.
 * \return Formatted error message with the list of errors with their location
 * in
 *        the parsed document. An empty string is returned if no error
 * occurred
 *        during parsing.
 * \deprecated Use getFormattedErrorMessages() instead (typo fix).
 */
JSONCPP_DEPRECATED("Use getFormattedErrorMessages() instead.")
std::string getFormatedErrorMessages() const;

/** \brief Returns a user friendly string that list errors in the parsed
 * document.
 * \return Formatted error message with the list of errors with their location
 * in
 *        the parsed document. An empty string is returned if no error
 * occurred
 *        during parsing.
 */
std::string getFormattedErrorMessages() const;

/** \brief Returns a vector of structured erros encounted while parsing.
 * \return A (possibly empty) vector of StructuredError objects. Currently
 *        only one error can be returned, but the caller should tolerate
 * multiple
 *        errors.  This can occur if the parser recovers from a non-fatal
 *        parse error and then encounters additional errors.
 */
std::vector<StructuredError> getStructuredErrors() const;

/** \brief Add a semantic error message.
 * \param value JSON Value location associated with the error
 * \param message The error message.
 * \return \c true if the error was successfully added, \c false if the
 * Value offset exceeds the document size.
 */
bool pushError(const Value& value, const std::string& message);

/** \brief Add a semantic error message with extra context.
 * \param value JSON Value location associated with the error
 * \param message The error message.
 * \param extra Additional JSON Value location to contextualize the error
 * \return \c true if the error was successfully added, \c false if either
 * Value offset exceeds the document size.
 */
bool pushError(const Value& value, const std::string& message, const Value& extra);

/** \brief Return whether there are any errors.
 * \return \c true if there are no errors to report \c false if
 * errors have occurred.
```

```
  */
 bool good() const;

private:
 enum TokenType {
   tokenEndOfStream = 0,
   tokenObjectBegin,
   tokenObjectEnd,
   tokenArrayBegin,
   tokenArrayEnd,
   tokenString,
   tokenNumber,
   tokenTrue,
   tokenFalse,
   tokenNull,
   tokenArraySeparator,
   tokenMemberSeparator,
   tokenComment,
   tokenError
 };

 class Token {
 public:
   TokenType type_;
   Location start_;
   Location end_;
 };

 class ErrorInfo {
 public:
   Token token_;
   std::string message_;
   Location extra_;
 };

 typedef std::deque<ErrorInfo> Errors;

 bool readToken(Token& token);
 void skipSpaces();
 bool match(Location pattern, int patternLength);
 bool readComment();
 bool readCStyleComment();
 bool readCppStyleComment();
 bool readString();
 void readNumber();
 bool readValue();
 bool readObject(Token& token);
 bool readArray(Token& token);
 bool decodeNumber(Token& token);
 bool decodeNumber(Token& token, Value& decoded);
 bool decodeString(Token& token);
 bool decodeString(Token& token, std::string& decoded);
 bool decodeDouble(Token& token);
 bool decodeDouble(Token& token, Value& decoded);
 bool decodeUnicodeCodePoint(Token& token,
                             Location& current,
                             Location end,
                             unsigned int& unicode);
 bool decodeUnicodeEscapeSequence(Token& token,
                                  Location& current,
                                  Location end,
                                  unsigned int& unicode);
 bool addError(const std::string& message, Token& token, Location extra = 0);
 bool recoverFromError(TokenType skipUntilToken);
 bool addErrorAndRecover(const std::string& message,
                         Token& token,
                         TokenType skipUntilToken);
 void skipUntilSpace();
 Value& currentValue();
 Char getNextChar();
 void
 getLocationLineAndColumn(Location location, int& line, int& column) const;
 std::string getLocationLineAndColumn(Location location) const;
 void addComment(Location begin, Location end, CommentPlacement placement);
 void skipCommentTokens(Token& token);

 typedef std::stack<Value*> Nodes;
 Nodes nodes_;
 Errors errors_;
 std::string document_;
 Location begin_;
 Location end_;
 Location current_;
 Location lastValueEnd_;
 Value* lastValue_;
 std::string commentsBefore_;
 Features features_;
 bool collectComments_;
}; // Reader

/** Interface for reading JSON from a char array.
*/
class JSON_API CharReader {
public:
 virtual ~CharReader() {}
 /** \brief Read a Value from a <a HREF="http://www.json.org">JSON</a>
```

```
   document.
   * The document must be a UTF-8 encoded string containing the document to read.
   *
   * \param beginDoc Pointer on the beginning of the UTF-8 encoded string of the
   document to read.
   * \param endDoc Pointer on the end of the UTF-8 encoded string of the
   document to read.
   *        Must be >= beginDoc.
   * \param root [out] Contains the root value of the document if it was
   *             successfully parsed.
   * \param errs [out] Formatted error messages (if not NULL)
   *        a user friendly string that lists errors in the parsed
   * document.
   * \return \c true if the document was successfully parsed, \c false if an
   error occurred.
   */
  virtual bool parse(
      char const* beginDoc, char const* endDoc,
      Value* root, std::string* errs) = 0;

  class JSON_API Factory {
  public:
    virtual ~Factory() {}
    /** \brief Allocate a CharReader via operator new().
     * \throw std::exception if something goes wrong (e.g. invalid settings)
     */
    virtual CharReader* newCharReader() const = 0;
  };  // Factory
};  // CharReader

/** \brief Build a CharReader implementation.

Usage:
\code
  using namespace Json;
  CharReaderBuilder builder;
  builder["collectComments"] = false;
  Value value;
  std::string errs;
  bool ok = parseFromStream(builder, std::cin, &value, &errs);
\endcode
*/
class JSON_API CharReaderBuilder : public CharReader::Factory {
public:
  // Note: We use a Json::Value so that we can add data-members to this class
  // without a major version bump.
  /** Configuration of this builder.
    These are case-sensitive.
    Available settings (case-sensitive):
    - `"collectComments": false or true`
      - true to collect comment and allow writing them
        back during serialization, false to discard comments.
        This parameter is ignored if allowComments is false.
    - `"allowComments": false or true`
      - true if comments are allowed.
    - `"strictRoot": false or true`
      - true if root must be either an array or an object value
    - `"allowDroppedNullPlaceholders": false or true`
      - true if dropped null placeholders are allowed. (See StreamWriterBuilder.)
    - `"allowNumericKeys": false or true`
      - true if numeric object keys are allowed.
    - `"allowSingleQuotes": false or true`
      - true if '' are allowed for strings (both keys and values)
    - `"stackLimit": integer`
      - Exceeding stackLimit (recursive depth of `readValue()`) will
        cause an exception.
      - This is a security issue (seg-faults caused by deeply nested JSON),
        so the default is low.
    - `"failIfExtra": false or true`
      - If true, `parse()` returns false when extra non-whitespace trails
        the JSON value in the input string.
    - `"rejectDupKeys": false or true`
      - If true, `parse()` returns false when a key is duplicated within an object.
    - `"allowSpecialFloats": false or true`
      - If true, special float values (NaNs and infinities) are allowed
        and their values are lossfree restorable.

    You can examine 'settings_` yourself
    to see the defaults. You can also write and read them just like any
    JSON Value.
    \sa setDefaults()
    */
  Json::Value settings_;

  CharReaderBuilder();
  ~CharReaderBuilder() override;

  CharReader* newCharReader() const override;

  /** \return true if 'settings' are legal and consistent;
   *   otherwise, indicate bad settings via 'invalid'.
   */
  bool validate(Json::Value* invalid) const;

  /** A simple way to update a specific setting.
   */
  Value& operator[](std::string key);

  /** Called by ctor, but you can use this to reset settings_.
   * \pre 'settings' != NULL (but Json::null is fine)
   * \remark Defaults:
   * \snippet src/lib_json/json_reader.cpp CharReaderBuilderDefaults
   */
  static void setDefaults(Json::Value* settings);
  /** Same as old Features::strictMode().
   * \pre 'settings' != NULL (but Json::null is fine)
   * \remark Defaults:
   * \snippet src/lib_json/json_reader.cpp CharReaderBuilderStrictMode
   */
  static void strictMode(Json::Value* settings);
 };

 /** Consume entire stream and use its begin/end.
  * Someday we might have a real StreamReader, but for now this
  * is convenient.
  */
 bool JSON_API parseFromStream(
    CharReader::Factory const&,
    std::istream&,
    Value* root, std::string* errs);

 /** \brief Read from 'sin' into 'root'.

 Always keep comments from the input JSON.

 This can be used to read a file into a particular sub-object.
 For example:
 \code
 Json::Value root;
 cin >> root["dir"]["file"];
 cout << root;
 \endcode
 Result:
 \verbatim
 {
 "dir": {
     "file": {
     // The input stream JSON would be nested here.
     }
 }
 }
 \endverbatim
 \throw std::exception on parse error.
 \see Json::operator<<()
```

```
*/
JSON_API std::istream& operator>>(std::istream&, Value&);

} // namespace Json

#if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)
#pragma warning(pop)
#endif // if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)

#endif // CPPTL_JSON_READER_H_INCLUDED

// //////////////////////////////////////////////////////////////////////
// End of content of file: include/json/reader.h
// //////////////////////////////////////////////////////////////////////






// //////////////////////////////////////////////////////////////////////
// Beginning of content of file: include/json/writer.h
// //////////////////////////////////////////////////////////////////////

// Copyright 2007-2010 Baptiste Lepilleur
// Distributed under MIT license, or public domain if desired and
// recognized in your jurisdiction.
// See file LICENSE for detail or copy at http://jsoncpp.sourceforge.net/LICENSE

#ifndef JSON_WRITER_H_INCLUDED
#define JSON_WRITER_H_INCLUDED

#if !defined(JSON_IS_AMALGAMATION)
#include "value.h"
#endif // if !defined(JSON_IS_AMALGAMATION)
#include <vector>
#include <string>
#include <ostream>

// Disable warning C4251: <data member>: <type> needs to have dll-interface to
// be used by...
#if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)
#pragma warning(push)
#pragma warning(disable : 4251)
#endif // if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)

namespace Json {
```

```
class Value;

/**

Usage:
\code
 using namespace Json;
 void writeToStdout(StreamWriter::Factory const& factory, Value const& value) {
   std::unique_ptr<StreamWriter> const writer(
     factory.newStreamWriter());
   writer->write(value, &std::cout);
   std::cout << std::endl;  // add lf and flush
 }
\endcode
*/
class JSON_API StreamWriter {
protected:
 std::ostream* sout_;  // not owned; will not delete
public:
 StreamWriter();
 virtual ~StreamWriter();
 /** Write Value into document as configured in sub-class.
     Do not take ownership of sout, but maintain a reference during function.
     \pre sout != NULL
     \return zero on success (For now, we always return zero, so check the stream instead.)
     \throw std::exception possibly, depending on configuration
  */
 virtual int write(Value const& root, std::ostream* sout) = 0;

 /** \brief A simple abstract factory.
  */
 class JSON_API Factory {
 public:
   virtual ~Factory();
   /** \brief Allocate a CharReader via operator new().
    * \throw std::exception if something goes wrong (e.g. invalid settings)
    */
   virtual StreamWriter* newStreamWriter() const = 0;
 };  // Factory
};  // StreamWriter

/** \brief Write into stringstream, then return string, for convenience.
* A StreamWriter will be created from the factory, used, and then deleted.
*/
std::string JSON_API writeString(StreamWriter::Factory const& factory, Value const& root);


/** \brief Build a StreamWriter implementation.
```

```
Usage:
\code
 using namespace Json;
 Value value = ...;
 StreamWriterBuilder builder;
 builder["commentStyle"] = "None";
 builder["indentation"] = "   ";  // or whatever you like
 std::unique_ptr<Json::StreamWriter> writer(
     builder.newStreamWriter());
 writer->write(value, &std::cout);
 std::cout << std::endl;  // add lf and flush
\endcode
*/
class JSON_API StreamWriterBuilder : public StreamWriter::Factory {
public:
 // Note: We use a Json::Value so that we can add data-members to this class
 // without a major version bump.
 /** Configuration of this builder.
   Available settings (case-sensitive):
   - "commentStyle": "None" or "All"
   - "indentation":  "<anything>"
   - "enableYAMLCompatibility": false or true
     - slightly change the whitespace around colons
   - "dropNullPlaceholders": false or true
     - Drop the "null" string from the writer's output for nullValues.
       Strictly speaking, this is not valid JSON. But when the output is being
       fed to a browser's Javascript, it makes for smaller output and the
       browser can handle the output just fine.
   - "useSpecialFloats": false or true
     - If true, outputs non-finite floating point values in the following way:
       NaN values as "NaN", positive infinity as "Infinity", and negative infinity
       as "-Infinity".

   You can examine 'settings_` yourself
   to see the defaults. You can also write and read them just like any
   JSON Value.
   \sa setDefaults()
   */
 Json::Value settings_;

 StreamWriterBuilder();
 ~StreamWriterBuilder() override;

 /**
  * \throw std::exception if something goes wrong (e.g. invalid settings)
  */
 StreamWriter* newStreamWriter() const override;
 /** \return true if 'settings' are legal and consistent;
  *   otherwise, indicate bad settings via 'invalid'.
  */
 bool validate(Json::Value* invalid) const;
 /** A simple way to update a specific setting.
  */
 Value& operator[](std::string key);

 /** Called by ctor, but you can use this to reset settings_.
  * \pre 'settings' != NULL (but Json::null is fine)
  * \remark Defaults:
  * \snippet src/lib_json/json_writer.cpp StreamWriterBuilderDefaults
  */
 static void setDefaults(Json::Value* settings);
};

/** \brief Abstract class for writers.
* \deprecated Use StreamWriter. (And really, this is an implementation detail.)
*/
class JSON_API Writer {
public:
 virtual ~Writer();

 virtual std::string write(const Value& root) = 0;
};

/** \brief Outputs a Value in <a HREF="http://www.json.org">JSON</a> format
*without formatting (not human friendly).
*
* The JSON document is written in a single line. It is not intended for 'human'
*consumption,
* but may be usefull to support feature such as RPC where bandwith is limited.
* \sa Reader, Value
* \deprecated Use StreamWriterBuilder.
*/
class JSON_API FastWriter : public Writer {

public:
 FastWriter();
 ~FastWriter() override {}

 void enableYAMLCompatibility();

 /** \brief Drop the "null" string from the writer's output for nullValues.
  * Strictly speaking, this is not valid JSON. But when the output is being
  * fed to a browser's Javascript, it makes for smaller output and the
  * browser can handle the output just fine.
  */
 void dropNullPlaceholders();

 void omitEndingLineFeed();

public: // overridden from Writer
 std::string write(const Value& root) override;

private:
 void writeValue(const Value& value);

 std::string document_;
 bool yamlCompatiblityEnabled_;
 bool dropNullPlaceholders_;
 bool omitEndingLineFeed_;
};

/** \brief Writes a Value in <a HREF="http://www.json.org">JSON</a> format in a
*human friendly way.
*
* The rules for line break and indent are as follow:
* - Object value:
*     - if empty then print {} without indent and line break
*     - if not empty the print '{', line break & indent, print one value per
*line
*       and then unindent and line break and print '}'.
* - Array value:
*     - if empty then print [] without indent and line break
*     - if the array contains no object value, empty array or some other value
*types,
*       and all the values fit on one lines, then print the array on a single
*line.
*     - otherwise, it the values do not fit on one line, or the array contains
*       object or non empty array, then print one value per line.
*
* If the Value have comments then they are outputed according to their
*#CommentPlacement.
*
* \sa Reader, Value, Value::setComment()
* \deprecated Use StreamWriterBuilder.
*/
class JSON_API StyledWriter : public Writer {
public:
 StyledWriter();
 ~StyledWriter() override {}

public: // overridden from Writer
 /** \brief Serialize a Value in <a HREF="http://www.json.org">JSON</a> format.
```

```
  * \param root Value to serialize.
  * \return String containing the JSON document that represents the root value.
  */
 std::string write(const Value& root) override;

private:
 void writeValue(const Value& value);
 void writeArrayValue(const Value& value);
 bool isMultineArray(const Value& value);
 void pushValue(const std::string& value);
 void writeIndent();
 void writeWithIndent(const std::string& value);
 void indent();
 void unindent();
 void writeCommentBeforeValue(const Value& root);
 void writeCommentAfterValueOnSameLine(const Value& root);
 bool hasCommentForValue(const Value& value);
 static std::string normalizeEOL(const std::string& text);

 typedef std::vector<std::string> ChildValues;

 ChildValues childValues_;
 std::string document_;
 std::string indentString_;
 int rightMargin_;
 int indentSize_;
 bool addChildValues_;
};

/** \brief Writes a Value in <a HREF="http://www.json.org">JSON</a> format in a
human friendly way,
    to a stream rather than to a string.
*
* The rules for line break and indent are as follow:
* - Object value:
*     - if empty then print {} without indent and line break
*     - if not empty the print '{', line break & indent, print one value per
line
*       and then unindent and line break and print '}'.
* - Array value:
*     - if empty then print [] without indent and line break
*     - if the array contains no object value, empty array or some other value
types,
*       and all the values fit on one lines, then print the array on a single
line.
*     - otherwise, it the values do not fit on one line, or the array contains
*       object or non empty array, then print one value per line.
*
```

```
* If the Value have comments then they are outputed according to their
#CommentPlacement.
*
* \param indentation Each level will be indented by this amount extra.
* \sa Reader, Value, Value::setComment()
* \deprecated Use StreamWriterBuilder.
*/
class JSON_API StyledStreamWriter {
public:
 StyledStreamWriter(std::string indentation = "\t");
 ~StyledStreamWriter() {}

public:
 /** \brief Serialize a Value in <a HREF="http://www.json.org">JSON</a> format.
  * \param out Stream to write to. (Can be ostringstream, e.g.)
  * \param root Value to serialize.
  * \note There is no point in deriving from Writer, since write() should not
  * return a value.
  */
 void write(std::ostream& out, const Value& root);

private:
 void writeValue(const Value& value);
 void writeArrayValue(const Value& value);
 bool isMultineArray(const Value& value);
 void pushValue(const std::string& value);
 void writeIndent();
 void writeWithIndent(const std::string& value);
 void indent();
 void unindent();
 void writeCommentBeforeValue(const Value& root);
 void writeCommentAfterValueOnSameLine(const Value& root);
 bool hasCommentForValue(const Value& value);
 static std::string normalizeEOL(const std::string& text);

 typedef std::vector<std::string> ChildValues;

 ChildValues childValues_;
 std::ostream* document_;
 std::string indentString_;
 int rightMargin_;
 std::string indentation_;
 bool addChildValues_ : 1;
 bool indented_ : 1;
};

#if defined(JSON_HAS_INT64)
std::string JSON_API valueToString(Int value);
```

```
std::string JSON_API valueToString(UInt value);
#endif // if defined(JSON_HAS_INT64)
std::string JSON_API valueToString(LargestInt value);
std::string JSON_API valueToString(LargestUInt value);
std::string JSON_API valueToString(double value);
std::string JSON_API valueToString(bool value);
std::string JSON_API valueToQuotedString(const char* value);

/// \brief Output using the StyledStreamWriter.
/// \see Json::operator>>()
JSON_API std::ostream& operator<<(std::ostream&, const Value& root);

} // namespace Json

#if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)
#pragma warning(pop)
#endif // if defined(JSONCPP_DISABLE_DLL_INTERFACE_WARNING)

#endif // JSON_WRITER_H_INCLUDED

// //////////////////////////////////////////////////////////////////////
// End of content of file: include/json/writer.h
// //////////////////////////////////////////////////////////////////////






// //////////////////////////////////////////////////////////////////////
// Beginning of content of file: include/json/assertions.h
// //////////////////////////////////////////////////////////////////////

// Copyright 2007-2010 Baptiste Lepilleur
// Distributed under MIT license, or public domain if desired and
// recognized in your jurisdiction.
// See file LICENSE for detail or copy at http://jsoncpp.sourceforge.net/LICENSE

#ifndef CPPTL_JSON_ASSERTIONS_H_INCLUDED
#define CPPTL_JSON_ASSERTIONS_H_INCLUDED

#include <stdlib.h>
#include <sstream>

#if !defined(JSON_IS_AMALGAMATION)
#include "config.h"
#endif // if !defined(JSON_IS_AMALGAMATION)
```

```
/** It should not be possible for a maliciously designed file to
*  cause an abort() or seg-fault, so these macros are used only
*  for pre-condition violations and internal logic errors.
*/
#if JSON_USE_EXCEPTION

// @todo <= add detail about condition in exception
# define JSON_ASSERT(condition)                                \
 {if (!(condition)) {Json::throwLogicError( "assert json failed" );}}

# define JSON_FAIL_MESSAGE(message)                                  \
 {                                                            \
   std::ostringstream oss; oss << message;                           \
   Json::throwLogicError(oss.str());                               \
   abort();                                                   \
 }

#else // JSON_USE_EXCEPTION

# define JSON_ASSERT(condition) assert(condition)

// The call to assert() will show the failure message in debug builds. In
// release builds we abort, for a core-dump or debugger.
# define JSON_FAIL_MESSAGE(message)                                  \
 {                                                            \
   std::ostringstream oss; oss << message;                           \
   assert(false && oss.str().c_str());                            \
   abort();                                                   \
 }


#endif

#define JSON_ASSERT_MESSAGE(condition, message)                          \
 if (!(condition)) {                                              \
   JSON_FAIL_MESSAGE(message);                                      \
 }

#endif // CPPTL_JSON_ASSERTIONS_H_INCLUDED

// //////////////////////////////////////////////////////////////////////
// End of content of file: include/json/assertions.h
// //////////////////////////////////////////////////////////////////////
```

```
#endif //ifndef JSON_AMALGATED_H_INCLUDED
```

```
This file contains a list of people who have made large contributions
to the public version of Protocol Buffers.

Original Protocol Buffers design and implementation:
 Sanjay Ghemawat <sanjay@google.com>
 Jeff Dean <jeff@google.com>
 Daniel Dulitz <daniel@google.com>
 Craig Silverstein
 Paul Haahr <haahr@google.com>
 Corey Anderson <corin@google.com>
 (and many others)

Proto2 C++ and Java primary author:
 Kenton Varda <kenton@google.com>

Proto2 Python primary authors:
 Will Robinson <robinson@google.com>
 Petar Petrov <petar@google.com>

Java Nano primary authors:
 Brian Duff <bduff@google.com>
 Tom Chao <chaot@google.com>
 Max Cai <maxtroy@google.com>
 Ulas Kirazci <ulas@google.com>

Large code contributions:
 Jason Hsueh <jasonh@google.com>
 Joseph Schorr <jschorr@google.com>
 Wenbo Zhu <wenboz@google.com>

Large quantity of code reviews:
 Scott Bruce <sbruce@google.com>
 Frank Yellin
 Neal Norwitz <nnorwitz@google.com>
 Jeffrey Yasskin <jyasskin@google.com>
 Ambrose Feinstein <ambrose@google.com>

Documentation:
 Lisa Carey <lcarey@google.com>

Maven packaging:
 Gregory Kick <gak@google.com>

Patch contributors:
 Kevin Ko <kevin.s.ko@gmail.com>
   * Small patch to handle trailing slashes in --proto_path flag.
 Johan Euphrosine <proppy@aminche.com>
```

```
  * Small patch to fix Python CallMethod().
Ulrich Kunitz <kune@deine-taler.de>
  * Small optimizations to Python serialization.
Leandro Lucarella <llucax@gmail.com>
  * VI syntax highlighting tweaks.
  * Fix compiler to not make output executable.
Dilip Joseph <dilip.antony.joseph@gmail.com>
  * Heuristic detection of sub-messages when printing unknown fields in
    text format.
Brian Atkinson <nairb774@gmail.com>
  * Added @Override annotation to generated Java code where appropriate.
Vincent Choinire <Choiniere.Vincent@hydro.qc.ca>
  * Tru64 support.
Monty Taylor <monty.taylor@gmail.com>
  * Solaris 10 + Sun Studio fixes.
Alek Storm <alek.storm@gmail.com>
  * Slicing support for repeated scalar fields for the Python API.
Oleg Smolsky <oleg.smolsky@gmail.com>
  * MS Visual Studio error format option.
  * Detect unordered_map in stl_hash.m4.
Brian Olson <brianolson@google.com>
  * gzip/zlib I/O support.
Michael Poole <mdpoole@troilus.org>
  * Fixed warnings about generated constructors not explicitly initializing
    all fields (only present with certain compiler settings).
  * Added generation of field number constants.
Wink Saville <wink@google.com>
  * Fixed initialization ordering problem in logging code.
Will Pierce <willp@nuclei.com>
  * Small patch improving performance of in Python serialization.
Alexandre Vassalotti <alexandre@peadrop.com>
  * Emacs mode for Protocol Buffers (editors/protobuf-mode.el).
Scott Stafford <scott.stafford@gmail.com>
  * Added Swap(), SwapElements(), and RemoveLast() to Reflection interface.
Alexander Melnikov <alm@sibmail.ru>
  * HPUX support.
Oliver Jowett <oliver.jowett@gmail.com>
  * Detect whether zlib is new enough in configure script.
  * Fixes for Solaris 10 32/64-bit confusion.
Evan Jones <evanj@mit.edu>
  * Optimize Java serialization code when writing a small message to a stream.
  * Optimize Java serialization of strings so that UTF-8 encoding happens only
    once per string per serialization call.
  * Clean up some Java warnings.
  * Fix bug with permanent callbacks that delete themselves when run.
Michael Kucharski <m.kucharski@gmail.com>
  * Added CodedInputStream.getTotalBytesRead().
Kacper Kowalik <xarthisius.kk@gmail.com>
```

  * Fixed m4/acx_pthread.m4 problem for some Linux distributions.
William Orr <will@worrbase.com>
  * Fixed detection of sched_yield on Solaris.
  * Added atomicops for Solaris
Andrew Paprocki <andrew@ishiboo.com>
  * Fixed minor IBM xlC compiler build issues
  * Added atomicops for AIX (POWER)

1.8 commons-io 2.5

1.8.1 Available under license :

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache Commons IO
Copyright 2002-2016 The Apache Software Foundation

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

1.9 commons-lang3 3.0

1.9.1 Available under license :

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache Commons Lang

Copyright 2001-2011 The Apache Software Foundation

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

This product includes software from the Spring Framework,

under the Apache License 2.0 (see: StringUtils.containsWhitespace())

1.10 confluent-kafka-go 0.11.0

1.10.1 Available under license :

Apache License

Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

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

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

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

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

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

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

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright {yyyy} {name of copyright owner}

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

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

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

1.11 d3 5.0.0

1.11.1 Available under license :

Copyright 2010-2017 Mike Bostock
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 author nor the names of contributors may be used to
 endorse or promote products derived from this software without specific prior
 written permission.

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

1.12 disruptor 3.3.4

1.12.1 Available under license :

The Apache Software License, Version 2.0
http://www.apache.org/licenses/LICENSE-2.0.txt

1.13 dlintw/goconf master

1.13.1 Available under license :

Copyright (c) 2010, Stephen Weinberg
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 goconf 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.

1.14 eapache/queue master

1.14.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Evan Huus

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.

1.15 elasticsearch 6.3.1

1.15.1 Available under license :

Mozilla Public License Version 2.0
==================================

1. Definitions

1.1. "Contributor"
means each individual or legal entity that creates, contributes to
the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used
by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached
the notice in Exhibit A, the Executable Form of such Source Code
Form, and Modifications of such Source Code Form, in each case
including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described
in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of

version 1.1 or earlier of the License, but not also under the
terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in
a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible,
whether at the time of the initial grant or subsequently, any and
all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to,
deletion from, or modification of the contents of Covered
Software; or

(b) any new file in Source Code Form that contains any Covered
Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method,
process, and apparatus claims, in any patent Licensable by such
Contributor that would be infringed, but for the grant of the
License, by the making, using, selling, offering for sale, having
made, import, or transfer of either its Contributions or its
Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU
Lesser General Public License, Version 2.1, the GNU Affero General
Public License, Version 3.0, or any later versions of those
licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
means an individual or a legal entity exercising rights under this
License. For legal entities, "You" includes any entity that

controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code

Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this

License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*                                                                      *
*  Under no circumstances and under no legal theory, whether tort      *
*  (including negligence), contract, or otherwise, shall any           *
*  Contributor, or anyone who distributes Covered Software as          *
*  permitted above, be liable to You for any direct, indirect,         *
*  special, incidental, or consequential damages of any character      *
*  including, without limitation, damages for lost profits, loss of    *
*  goodwill, work stoppage, computer failure or malfunction, or any    *
*  and all other commercial damages or losses, even if such party      *
*  shall have been informed of the possibility of such damages. This   *
*  limitation of liability shall not apply to liability for death or   *
*  personal injury resulting from such party's negligence to the       *
*  extent applicable law prohibits such limitation. Some               *
*  jurisdictions do not allow the exclusion or limitation of           *
*  incidental or consequential damages, so this exclusion and          *
*  limitation may not apply to You.                                    *
*                                                                      *
************************************************************************
```

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as
defined by the Mozilla Public License, v. 2.0.
This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
Except where otherwise noted in the source code (e.g. the files hash.c,
list.c and the trio files, which are covered by a similar licence but
with different Copyright notices) all the files are:

Copyright (C) 1998-2003 Daniel Veillard. All Rights Reserved.

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD 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 Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.
name,version,revision,url,license,copyright
Visual Studio 2013 Express Runtime Libraries,12.0,,https://www.visualstudio.com,Custom;https://s3-eu-west-1.amazonaws.com/prelert-artifacts/dependencies/License_Visual_Studio_2013_Express_English_b8a9dbc1-1fd4-4103-a83b-a2896f193ea0.htm,Copyright (C) Microsoft Corporation
Copyright (C) 1995-2013 Jean-loup Gailly and Mark Adler

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.

Jean-loup Gailly   Mark Adler
jloup@gzip.org   madler@alumni.caltech.edu

name,version,revision,url,license,copyright
RapidXml,1.13,,http://rapidxml.sourceforge.net,MIT,"Copyright (c) 2006, 2007 Marcin Kalicinski"
name,version,revision,url,license,copyright
Apache Portable Runtime Library,1.5.2,,https://apr.apache.org,Apache-2.0,Copyright (c) 2000-2015 The Apache Software Foundation

The Moby lexicon project is complete and has been place into the public domain. Use, sell, rework, excerpt and use in any way on any platform.

Placing this material on internal or public servers is also encouraged. The compiler is not aware of any export restrictions so freely distribute world-wide.

You can verify the public domain status by contacting

Grady Ward
3449 Martha Ct.
Arcata, CA 95521-4884

daedal@myrealbox.com >>

Tencent is pleased to support the open source community by making RapidJSON available.

Copyright (C) 2015 THL A29 Limited, a Tencent company, and Milo Yip.
All rights reserved.

Terms of the MIT License:

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.
Use of this software is granted under one of the following two licenses,
to be chosen freely by the user.

1. Boost Software License - Version 1.0 - August 17th, 2003
==

Copyright (c) 2006, 2007 Marcin Kalicinski

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

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, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2. The MIT License
==

Copyright (c) 2006, 2007 Marcin Kalicinski

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.

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Of the two options provided, Elastic chooses to license RapidXml under the
MIT License.

The collective work is Copyright 2000-2011 by Kevin Atkinson as well
as any of the copyrights mentioned below:

Copyright 2000-2011 by Kevin Atkinson

Permission to use, copy, modify, distribute and sell these word
lists, the associated scripts, the output created from the scripts,
and its documentation for any purpose is hereby granted without fee,
provided that the above copyright notice appears in all copies and
that both that copyright notice and this permission notice appear in
supporting documentation. Kevin Atkinson makes no representations
about the suitability of this array for any purpose. It is provided
"as is" without express or implied warranty.

Alan Beale <biljir@pobox.com> also deserves special credit as he has,
in addition to providing the 12Dicts package and being a major
contributor to the ENABLE word list, given me an incredible amount of
feedback and created a number of special lists (those found in the
Supplement) in order to help improve the overall quality of SCOWL.

The 10 level includes the 1000 most common English words (according to
the Moby (TM) Words II [MWords] package), a subset of the 1000 most
common words on the Internet (again, according to Moby Words II), and

frequently class 16 from Brian Kelk's "UK English Wordlist
with Frequency Classification".

The MWords package was explicitly placed in the public domain:

> The Moby lexicon project is complete and has
> been place into the public domain. Use, sell,
> rework, excerpt and use in any way on any platform.
>
> Placing this material on internal or public servers is
> also encouraged. The compiler is not aware of any
> export restrictions so freely distribute world-wide.
>
> You can verify the public domain status by contacting
>
> Grady Ward
> 3449 Martha Ct.
> Arcata, CA 95521-4884
>
> grady@netcom.com
> grady@northcoast.com

The "UK English Wordlist With Frequency Classification" is also in the
Public Domain:

Date: Sat, 08 Jul 2000 20:27:21 +0100
From: Brian Kelk <Brian.Kelk@cl.cam.ac.uk>

> I was wondering what the copyright status of your "UK English
> Wordlist With Frequency Classification" word list as it seems to
> be lacking any copyright notice.

There were many many sources in total, but any text marked
"copyright" was avoided. Locally-written documentation was one
source. An earlier version of the list resided in a filespace called
PUBLIC on the University mainframe, because it was considered public
domain.

Date: Tue, 11 Jul 2000 19:31:34 +0100

> So are you saying your word list is also in the public domain?

That is the intention.

The 20 level includes frequency classes 7-15 from Brian's word list.

The 35 level includes frequency classes 2-6 and words appearing in at
least 11 of 12 dictionaries as indicated in the 12Dicts package. All

words from the 12Dicts package have had likely inflections added via my inflection database.

The 12Dicts package and Supplement is in the Public Domain.

The WordNet database, which was used in the creation of the Inflections database, is under the following copyright:

This software and database is being provided to you, the LICENSEE, by Princeton University under the following license. By obtaining, using and/or copying this software and database, you agree that you have read, understood, and will comply with these terms and conditions.:

Permission to use, copy, modify and distribute this software and database and its documentation for any purpose and without fee or royalty is hereby granted, provided that you agree to comply with the following copyright notice and statements, including the disclaimer, and that the same appear on ALL copies of the software, database and documentation, including modifications that you make for internal use or for distribution.

WordNet 1.6 Copyright 1997 by Princeton University. All rights reserved.

THIS SOFTWARE AND DATABASE IS PROVIDED "AS IS" AND PRINCETON UNIVERSITY MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PRINCETON UNIVERSITY MAKES NO REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE LICENSED SOFTWARE, DATABASE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

The name of Princeton University or Princeton may not be used in advertising or publicity pertaining to distribution of the software and/or database. Title to copyright in this software, database and any associated documentation shall at all times remain with Princeton University and LICENSEE agrees to preserve same.

The 40 level includes words from Alan's 3esl list found in version 4.0 of his 12dicts package. Like his other stuff the 3esl list is also in the public domain.

The 50 level includes Brian's frequency class 1, words words appearing in at least 5 of 12 of the dictionaries as indicated in the 12Dicts package, and uppercase words in at least 4 of the previous 12 dictionaries. A decent number of proper names is also included: The top 1000 male, female, and Last names from the 1990 Census report; a

list of names sent to me by Alan Beale; and a few names that I added
myself. Finally a small list of abbreviations not commonly found in
other word lists is included.

The name files form the Census report is a government document which I
don't think can be copyrighted.

The file special-jargon.50 uses common.lst and word.lst from the
"Unofficial Jargon File Word Lists" which is derived from "The Jargon
File". All of which is in the Public Domain. This file also contain
a few extra UNIX terms which are found in the file "unix-terms" in the
special/ directory.

The 55 level includes words from Alan's 2of4brif list found in version
4.0 of his 12dicts package. Like his other stuff the 2of4brif is also
in the public domain.

The 60 level includes all words appearing in at least 2 of the 12
dictionaries as indicated by the 12Dicts package.

The 70 level includes Brian's frequency class 0 and the 74,550 common
dictionary words from the MWords package. The common dictionary words,
like those from the 12Dicts package, have had all likely inflections
added. The 70 level also included the 5desk list from version 4.0 of
the 12Dics package which is the public domain.

The 80 level includes the ENABLE word list, all the lists in the
ENABLE supplement package (except for ABLE), the "UK Advanced Cryptics
Dictionary" (UKACD), the list of signature words in from YAWL package,
and the 10,196 places list from the MWords package.

The ENABLE package, mainted by M\Cooper <thegrendel@theriver.com>,
is in the Public Domain:

The ENABLE master word list, WORD.LST, is herewith formally released
into the Public Domain. Anyone is free to use it or distribute it in
any manner they see fit. No fee or registration is required for its
use nor are "contributions" solicited (if you feel you absolutely
must contribute something for your own peace of mind, the authors of
the ENABLE list ask that you make a donation on their behalf to your
favorite charity). This word list is our gift to the Scrabble
community, as an alternate to "official" word lists. Game designers
may feel free to incorporate the WORD.LST into their games. Please
mention the source and credit us as originators of the list. Note
that if you, as a game designer, use the WORD.LST in your product,
you may still copyright and protect your product, but you may *not*
legally copyright or in any way restrict redistribution of the
WORD.LST portion of your product. This *may* under law restrict your

rights to restrict your users' rights, but that is only fair.

UKACD, by J Ross Beresford <ross@bryson.demon.co.uk>, is under the following copyright:

Copyright (c) J Ross Beresford 1993-1999. All Rights Reserved.

The following restriction is placed on the use of this publication: if The UK Advanced Cryptics Dictionary is used in a software package or redistributed in any form, the copyright notice must be prominently displayed and the text of this document must be included verbatim.

There are no other restrictions: I would like to see the list distributed as widely as possible.

The 95 level includes the 354,984 single words, 256,772 compound words, 4,946 female names and the 3,897 male names, and 21,986 names from the MWords package, ABLE.LST from the ENABLE Supplement, and some additional words found in my part-of-speech database that were not found anywhere else.

Accent information was taken from UKACD.

My VARCON package was used to create the American, British, and Canadian word list.

Since the original word lists used used in the VARCON package came from the Ispell distribution they are under the Ispell copyright:

Copyright 1993, Geoff Kuenning, Granada Hills, CA
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 modifications to the source code must be clearly marked as such. Binary redistributions based on modified source code must be clearly marked as modified versions in the documentation and/or other materials provided with the distribution.

(clause 4 removed with permission from Geoff Kuenning)

5. The name of Geoff Kuenning may not be used to endorse or promote

products derived from this software without specific prior
written permission.

THIS SOFTWARE IS PROVIDED BY GEOFF KUENNING 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 GEOFF KUENNING 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 Portable Runtime Utility Library
Copyright (c) 2009 The Apache Software Foundation.

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

Portions of this software were developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign.

This software contains code derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm, including various modifications by Spyglass Inc., Carnegie Mellon University, and Bell Communications Research, Inc (Bellcore).

name,version,revision,url,license,copyright
GCC Runtime Library,6.2.0,6ac74a62ba7258299cf85fbef9bf45333ddf10c0,https://gcc.gnu.org,GPL-3.0 WITH GCC-exception-3.1,"Copyright (C) 2016 Free Software Foundation, Inc."

The msinttypes r29
Copyright (c) 2006-2013 Alexander Chemeris
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 copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS''
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND 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.
name,version,revision,url,license,copyright
The msinttypes,r29,,https://code.google.com/archive/p/msinttypes/,BSD-3-Clause,Copyright (c) 2006-2013 Alexander Chemeris
name,version,revision,url,license,copyright
Boost C++ Libraries,1.65.1,,http://www.boost.org,BSL-1.0,
Copyright (c) 1997, 1998, 2005, 2008 The NetBSD Foundation, Inc.
All rights reserved.

This code was contributed to The NetBSD Foundation by Klaus Klein.
Heavily optimised by David Laight

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 NETBSD FOUNDATION, INC. 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 FOUNDATION 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.
name,version,revision,url,license,copyright
Libxml2,2.9.4,,http://xmlsoft.org/,MIT,Copyright (C) 1998-2012 Daniel Veillard
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization

obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

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, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
name,version,revision,url,license,copyright
Apache Portable Runtime Utility Library,1.5.4,,https://apr.apache.org,Apache-2.0,Copyright (c) 2000-2014 The Apache Software Foundation
name,version,revision,url,license,copyright
SCOWL,7.1,,http://wordlist.aspell.net,Custom;https://raw.githubusercontent.com/en-wl/wordlist/scowl-7.1/scowl/Copyright,Copyright 2000-2011 by Kevin Atkinson
name,version,revision,url,license,copyright
Eigen,3.2.5,bdd17ee3b1b3a166cd5ec36dcad4fc1f3faf774a,http://eigen.tuxfamily.org,MPL-2.0,
name,version,revision,url,license,copyright
RapidJSON,1.1.0,f05edc9296507a9864d99931e203631c2ffd8d4a,http://rapidjson.org,MIT,"Copyright (C) 2015 THL A29 Limited, a Tencent company, and Milo Yip"
name,version,revision,url,license,copyright
Apache Portable Runtime Iconv Library,1.2.1,,https://apr.apache.org,Apache-2.0,Copyright (c) 2000 Konstantin Chuguev
name,version,revision,url,license,copyright
strptime,1.35,,http://cvsweb.netbsd.org/bsdweb.cgi/src/lib/libc/time/strptime.c,BSD-2-Clause-NetBSD,"Copyright (c) 1997, 1998, 2005, 2008 The NetBSD Foundation, Inc."
name,version,revision,url,license,copyright
zlib,1.2.8,,http://zlib.net,Zlib,Copyright (C) 1995-2013 Jean-loup Gailly and Mark Adler
Apache Portable Runtime
Copyright (c) 2009 The Apache Software Foundation.

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

Portions of this software were developed at the National Center
for Supercomputing Applications (NCSA) at the University of
Illinois at Urbana-Champaign.

This software contains code derived from the RSA Data Security
Inc. MD5 Message-Digest Algorithm.

This software contains code derived from UNIX V7, Copyright(C)
Caldera International Inc.
name,version,revision,url,license,copyright
log4cxx,0.10.0,,https://logging.apache.org/log4cxx,Apache-2.0,Copyright 2004-2007 The Apache Software Foundation
name,version,revision,url,license,copyright
Moby lexicon project,1,,http://icon.shef.ac.uk/Moby/,Public-Domain;http://icon.shef.ac.uk/Moby/,
GCC RUNTIME LIBRARY EXCEPTION

Version 3.1, 31 March 2009

Copyright (C) 2009 Free Software Foundation, Inc. <http://fsf.org/>

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

This GCC Runtime Library Exception ("Exception") is an additional
permission under section 7 of the GNU General Public License, version
3 ("GPLv3"). It applies to a given file (the "Runtime Library") that
bears a notice placed by the copyright holder of the file stating that
the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of
certain GCC header files and runtime libraries with the compiled
program. The purpose of this Exception is to allow compilation of
non-GPL (including proprietary) programs to use, in this way, the
header files and runtime libraries covered by this Exception.

0. Definitions.

A file is an "Independent Module" if it either requires the Runtime
Library for execution after a Compilation Process, or makes use of an
interface provided by the Runtime Library, but is not otherwise based
on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without
modifications, governed by version 3 (or a specified later version) of
the GNU General Public License (GPL) with the option of using any
subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation,
modification and use would permit combination with GCC in accord with
the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual
target processor architecture, in executable form or suitable for
input to an assembler, loader, linker and/or execution
phase. Notwithstanding that, Target Code does not include data in any
format that is used as a compiler intermediate representation, or used
for producing a compiler intermediate representation.

The "Compilation Process" transforms code entirely represented in
non-intermediate languages designed for human-written code, and/or in
Java Virtual Machine byte code, into Target Code. Thus, for example,
use of source code generators and preprocessors need not be considered
part of the Compilation Process, since the Compilation Process can be
understood as starting with the output of the generators or
preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or
with other GPL-compatible software, or if it is done without using any
work based on GCC. For example, using non-GPL-compatible Software to
optimize any GCC intermediate representations would not qualify as an
Eligible Compilation Process.

1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by
combining the Runtime Library with Independent Modules, even if such
propagation would otherwise violate the terms of GPLv3, provided that
all Target Code was generated by Eligible Compilation Processes. You
may then convey such a combination under terms of your choice,
consistent with the licensing of the Independent Modules.

2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general
presumption that third-party software is unaffected by the copyleft
requirements of the license of GCC.
MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO EXPRESS 2013 FOR WINDOWS DESKTOP

These license terms are an agreement between Microsoft Corporation (or based
on where you live, one of its affiliates) and you. Please read them. They
apply to the software named above, which includes the media on which you
received it, if any. The terms also apply to any Microsoft

*	updates,
*	supplements,
*	Internet-based services, and
*	support services

for this software, unless other terms accompany those items. If so, those terms apply.

BY USING THE SOFTWARE, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT USE THE SOFTWARE.

AS DESCRIBED BELOW, USING SOME FEATURES ALSO OPERATES AS YOUR CONSENT TO THE TRANSMISSION OF CERTAIN STANDARD COMPUTER INFORMATION FOR INTERNET-BASED SERVICES.

If you comply with these license terms, you have the rights below.
1.   INSTALLATION AND USE RIGHTS.
a.   Installation and Use. One user may install and use any number of copies of the software on your devices to design, develop and test your programs.
b.   Trial Edition. The initial installation of the software is a trial edition. The length of the trial period is thirty (30) days. You may convert your trial rights at any time to the full-use rights described below by registering with Microsoft.   After the expiration of the 30-day trial period, without registering, the trial edition will stop running. You may not be able to access data used with the trial edition when it stops running.
c.   Included Microsoft Programs. The software contains other Microsoft programs. The license terms with those programs apply to your use of them.
d.   Third Party Notices. Additional copyright notices and license terms applicable to portions of the software are set forth in the ThirdPartyNotices file accompanying the Microsoft software. In addition to any terms and conditions of any third party license identified in the ThirdPartyNotices file, the disclaimer of warranty and limitation on and exclusion of remedies and damages provisions of this agreement shall apply to all software in this distribution.
e.   Additional Licensing Requirements and/or Distribution Rights for Features Made Available with the Software:
*   Windows Phone Store. If you intend to make your program available in the Windows Phone Store, you will need to comply with the terms and conditions of the Windows Phone Store Application Provider Agreement.
*   Bing Maps. The software may include features that retrieve content such as maps, images and other data through the Bing Maps (or successor branded) application programming interface (the "Bing Maps API") to create reports displaying data on top of maps, aerial and hybrid imagery. If these features are included, you may use these features to create and view dynamic or static documents only in conjunction with and through methods and means of access integrated in the software. You may not otherwise copy, store, archive, or create a database of the entity information including business names, addresses and geocodes available through the Bing Maps API You may not use the Bing Maps API to provide sensor based

guidance/routing, nor use any Road Traffic Data or Bird's Eye
Imagery (or associated metadata) even if available through the Bing
Maps API for any purpose. Your use of the Bing Maps API and
associated content is also subject to the additional terms and
conditions at http://go.microsoft.com/fwlink/?LinkId=21969.
*   Additional Mapping APIs. The software may include application
programming interfaces that provide maps and other related mapping
features and services that are not provided by Bing (the "Additional
Mapping APIs"). These Additional Mapping APIs are subject to
additional terms and conditions and may require payment of fees to
Microsoft and/or third party providers based on the use or volume of
use of such Additional Mapping APIs. These terms and conditions
will be provided when you obtain any necessary license keys to use
such Additional Mapping APIs or when you review or receive
documentation related to the use of such Additional Mapping APIs.
*   Push Notifications. The Microsoft Push Notification Service may
not be used to send notifications that are mission critical or
otherwise could affect matters of life or death, including without
limitation critical notifications related to a medical device or
condition. MICROSOFT EXPRESSLY DISCLAIMS ANY WARRANTIES THAT THE USE
OF THE MICROSOFT PUSH NOTIFICATION SERVICE OR DELIVERY OF MICROSOFT
PUSH NOTIFICATION SERVICE NOTIFICATIONS WILL BE UNINTERRUPTED, ERROR
FREE, OR OTHERWISE GUARANTEED TO OCCUR ON A REAL-TIME BASIS.
*   Windows.Phone.Speech namespace API. Using speech recognition
functionality via the Windows.Phone.Speech namespace APIs in a
program requires the support of a speech recognition service. The
service may require network connectivity at the time of recognition
(e.g., when using a predefined grammar). In addition, the service
may also collect speech-related data in order to provide and improve
the service. The speech-related data may include, for example,
information related to grammar size and string phrases in a grammar.
Also, in order for a user to use speech recognition on the phone
they must first accept certain terms of use. The terms of use notify
the user that data related to their use of the speech recognition
service will be collected and used to provide and improve the
service. If a user does not accept the terms of use and speech
recognition is attempted by the application, the operation will not
work and an error will be returned to the application.
*   Location Framework. The software may contain a location
framework component that enables support of location services in
programs. In addition to the other limitations in this agreement,
you must comply with all applicable local laws and regulations when
using the location framework component or the rest of the software.
*   Device ID Access. The software may contain a component that
enables programs to access the device ID of the device that is
running the program. In addition to the other limitations in this
agreement, you must comply with all applicable local laws and
regulations when using the device ID access component or the rest of

the software.
*   PlayReady Support. The software includes the Windows Phone Emulator, which contains Microsoft's PlayReady content access technology. Content owners use Microsoft PlayReady content access technology to help protect their intellectual property, including copyrighted content. This software uses PlayReady technology to access PlayReady-protected content and/or Microsoft Digital Rights Management (DRM) protected content. Microsoft may decide to revoke the software's ability to consume PlayReady-protected content for reasons including but not limited to (i) if a breach or potential breach of PlayReady technology occurs, (ii) proactive robustness enhancement, and (iii) if content owners require the revocation because the software fails to properly enforce restrictions on content usage. Revocation should not affect unprotected content or content protected by other content access technologies. Content owners may require you to upgrade PlayReady to access their content. If you decline an upgrade, you will not be able to access content that requires the upgrade and may not be able to install other operating system updates or upgrades.
*   Microsoft Advertising SDKs. If you pass precise location data or other user related data (e.g., user identifier, profile data, behaviorally tracked user data) to the Microsoft Advertising SDK for Windows and Windows Phone or the Microsoft Advertising SDK, then your program must (a) notify end users that it will be collecting and using user related information and providing this information to Microsoft for Microsoft's advertising use, and (b) explicitly obtain affirmative user consent (e.g. the user must click an "Accept" or continue "Install" button) for this upon download of the software and/or application. In addition, you agree to: (a) comply with certification and other requirements for Windows and Windows Phone and Windows Store; (b) comply with Microsoft's privacy and other policies in your collection and use of any user data; (c) not collect or use any user identifier created or provided to you by Microsoft for any purpose other than passing such identifier to a Microsoft advertising service as part of your use of the service; and (d) provide in your privacy policy and/or terms of use a link that provides users with the ability to opt out of Microsoft interest based advertising at the following location https://choice.live.com/AdvertisementChoice/.
2.   ADDITIONAL LICENSING REQUIREMENTS AND/OR USE RIGHTS.
a.   Distributable Code. The software contains code that you are permitted to distribute in programs you develop if you comply with the terms below. (For this Section the term "distribution" also means deployment of your programs for third parties to access over the Internet.)
i.   Right to Use and Distribute. The code and text files listed below are "Distributable Code."
*   REDIST.TXT Files. You may copy and distribute the object code form of code listed on the REDIST list located at

go.microsoft.com/fwlink/?LinkId=313603.
* Sample Code You may modify, copy and distribute the source and
object code form of code marked as "sample."
* Third Party Distribution. You may permit distributors of your
programs to copy and distribute the Distributable Code as part of
those programs.
ii. Distribution Requirements. For any Distributable Code you
distribute, you must
* add significant primary functionality to it in your
programs;
* for any Distributable Code having a filename extension of .lib,
distribute only the results of running such Distributable Code
through a linker with your program;
* distribute Distributable Code included in a setup program only
as part of that setup program without modification;
* require distributors and external end users to agree to
terms that protect it at least as much as this agreement;
* display your valid copyright notice on your programs; and
* indemnify, defend, and hold harmless Microsoft from any
claims, including attorneys' fees, related to the distribution or
use of your programs.
iii. Distribution Restrictions. You may not
* alter any copyright, trademark or patent notice in the
Distributable Code;
* use Microsoft's trademarks in your programs' names or in a
way that suggests your programs come from or are endorsed by
Microsoft;
* distribute Distributable Code to run on a platform other
than Microsoft operating systems, run-time technologies, or
application platforms;
* include Distributable Code in malicious, deceptive or
unlawful programs; or
* modify or distribute the source code of any Distributable
Code so that any part of it becomes subject to an Excluded
License. An Excluded License is one that requires, as a condition
of use, modification or distribution, that
* the code be disclosed or distributed in source code form;
or
* others have the right to modify it.
3. DATA.
a. Consent for Internet-Based Services. The software features described
below and in the Visual Studio 2013 privacy statement connect to
Microsoft or service provider computer systems over the Internet. In
some cases, you will not receive a separate notice when they connect.
You may switch off these features or not use them. For more information
about these features, see http://go.microsoft.com/fwlink/?LinkId=286720
. BY USING THESE FEATURES, YOU CONSENT TO THE TRANSMISSION OF THIS
INFORMATION. Microsoft does not use the information to identify or

contact you.

* Extensions and Updates; and New Project Dialog. Extensions and Updates; and New Project Dialog can retrieve other software through the Internet from Visual Studio Gallery, Microsoft Download Center and MSDN Samples websites. To provide this other software, Extensions and Updates and; and New Project Dialog send to Microsoft the name and version of the software you are using and language code of the device where you installed the software. In addition, Extensions and Updates also contains an automatic update feature that is on by default.

? For more information about this feature, including instructions for to turning it off, see http://go.microsoft.com/fwlink/?LinkId=286720 . You may turn off this automatic update feature while the software is running ("opt out"). Unless you expressly opt out of this feature, the feature will (a) connect to Microsoft or service provider computer systems over the Internet, (b) use Internet protocols to send to the appropriate systems standard computer information, such as your computer's Internet protocol address, operating system, the name and version of the software you are using and language code of the device where you installed the software, and (c) prompt you to download and/or install, current updates to the software or other third party software. In some cases, you will not receive a separate notice before this feature takes effect. By installing the software, you consent to the transmission of standard computer information.

* Notification Tool Window. The software includes a notification tool window. When connected to the Internet, technologies that use the notification tool window may provide information related to extensions and other software or services that you have installed or opted into and display those updates to you. For more information please see the privacy statement http://go.microsoft.com/fwlink/?LinkId=286720 .

* Digital Certificates. The software uses digital certificates. These digital certificates confirm the identity of Internet users sending X.509 standard encrypted information. They also can be used to digitally sign files and macros to verify the integrity and origin of the file contents. The software retrieves certificates and updates certificate revocation lists using the Internet, when available.

* Really Simple Syndication ("RSS") Feed. This software start page contains updated content that is supplied by means of an RSS feed online from Microsoft.

* Package Manager Features. For more information about these features, see the privacy statement available at go.microsoft.com/fwlink/?LinkId=205205. With respect to these package manager features, we or the third party service provider may

use the computer information obtained through your use of these features, to improve our or their software and services. We or they may also share it with others, such as hardware and software vendors. They may use the information to improve how their products run with Microsoft software.

o Open Data Protocol (OData) Service. The Microsoft NuGet-Based Package Manager and the package manager feature of Microsoft ASP.NET Web Pages will access a list of packages that is supplied by means of an OData service online from Microsoft or a third-party service provider, located at go.microsoft.com/fwlink/?LinkID=206669. You may change the feed URL that the package manager feature initially points to at any time at your discretion.

Misuse of Internet-based Services. You may not use these services in any way that could harm them or impair anyone else's use of them. You may not use the services to try to gain unauthorized access to any service, data, account or network by any means.

b.   Computer Information. The following features use Internet protocols, which send to the appropriate systems computer information, such as your Internet protocol address, the type of operating system, browser and name and version of the software you are using, and the language code of the device where you installed the software. Microsoft uses this information to make the Internet-based services available to you.

*   Visual Studio Experience Improvement Program. This software uses the Visual Studio Experience Improvement Program ("VSEIP") which is on by default. VSEIP helps Microsoft collect information about problems that you have while using the software. Microsoft also uses VSEIP Information to improve its software and services. Microsoft may also share VSEIP data with others, such as agents acting on Microsoft's behalf and hardware and software vendors to improve how their products run with Microsoft software. To learn more about VSEIP, including how you can opt out, see go.microsoft.com/fwlink/?LinkId=286720 .

*   Data Collected Automatically. Microsoft automatically collects information identifying your installed Microsoft product, the operating system of the device, the CPU architecture of the operating system and data regarding the success or failure of the installation of the software, data identifying the cause of a crash in the product and information about the product license which is in use. Microsoft does not use this information to identify or contact you. For more information about privacy, see http://go.microsoft.com/fwlink/?LinkId=286720 .

c.   Microsoft Accounts in Visual Studio. If you are running the software on Windows 8, Windows 7 with sign-in assistant, or any other version of Windows that supports providing a Microsoft account directly to the software and you are signed into a Microsoft account in those versions of Windows, you may automatically be signed into the software and Visual Studio services accessed by the software using the same Microsoft account. This allows you to access services within the

software and roam the software's settings without being asked to
reenter your Microsoft account credentials each time you start the
software. For more information about signing into the software and the
services available therein with a Microsoft Account, see the privacy
statement http://go.microsoft.com/fwlink/?LinkId=286720 .

d. Use of Information. Microsoft may use the computer and services
information to improve its software and services. Microsoft may also
share it with others, such as hardware and software vendors. They may
use the information to improve how their products run with Microsoft
software.

4. Windows Components. The software may include Microsoft .NET Framework
software, certain .dll's related to Microsoft Build technologies and packaging
technologies, Windows controls, resource resolution and rendering
technologies, Windows Identity Foundation, Windows Web Services (WWS) for
Windows 2008 and Windows Library for JavaScript components. These software
components are part of Windows and are licensed to you under separate Windows
license terms or as a supplement to a Windows license. The product lifecycle
support and license terms for the applicable Windows product apply to your use
of these Windows components.

5. Microsoft .NET Framework Benchmark Testing. The software includes one or
more components of the .NET Framework (".NET Components"). You may conduct
internal benchmark testing of those components. You may disclose the results
of any benchmark test of those components, provided that you comply with the
conditions set forth at http://go.microsoft.com/fwlink/?LinkID=66406.
Notwithstanding any other agreement you may have with Microsoft, if you
disclose such benchmark test results, Microsoft shall have the right to
disclose the results of benchmark tests it conducts of your products that
compete with the applicable .NET Component, provided it complies with the same
conditions set forth at http://go.microsoft.com/fwlink/?LinkID=66406.

6. SQL SERVER SOFTWARE AND WINDOWS SOFTWARE DEVELOPMENT KIT (WINDOWS
SDK). The software is also accompanied by Microsoft SQL Server Software
and Windows SDK which are licensed to you under their own terms. The
license terms for the SQL Server Software and Windows SDK are located in
the "Licenses" folder of the software installation directory. If you do
not agree to the components' license terms, you may not use them.

7. PACKAGE MANAGER AND THIRD PARTY SOFTWARE INSTALLATION FEATURES. The software
includes the following features (each a "Feature"), each of which enables you
to obtain software applications or packages through the Internet from other
sources: Extension Manager, New Project Dialog, Web Platform Installer, and
Microsoft NuGet-Based Package Manager. Those software applications and
packages are offered and distributed in some cases by third parties and in
some cases by Microsoft, but each such application or package is under its own
license terms. Microsoft is not developing, distributing or licensing any of
the third-party applications or packages to you, but instead, as a
convenience, enables you to use the Features to access or obtain those
applications or packages directly from the third-party application or package
providers. By using the Features, you acknowledge and agree that:

* you are obtaining the applications or packages from such third

parties and under separate license terms applicable to each application
or package (including, with respect to the package-manager Features, any
terms applicable to software dependencies that may be included in the
package);
MICROSOFT MAKES NO REPRESENTATIONS, WARRANTIES OR GUARANTEES AS TO THE FEED OR GALLERY URL, ANY FEEDS OR GALLERIES FROM SUCH URL, THE INFORMATION CONTAINED THEREIN, OR ANY SOFTWARE APPLICATIONS OR PACKAGES REFERENCED IN OR ACCESSED BY YOU THROUGH SUCH FEEDS OR GALLERIES. MICROSOFT GRANTS YOU NO LICENSE RIGHTS FOR THIRD-PARTY SOFTWARE APPLICATIONS OR PACKAGES THAT ARE OBTAINED USING THE FEATURES.

8.   SCOPE OF LICENSE. The software is licensed, not sold. This agreement only gives you some rights to use the software. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that only allow you to use it in certain ways. You may not

*   disclose the results of any benchmark tests of the software to any third party without Microsoft's prior written approval, however, this does not apply to the Microsoft .NET Framework (see Section 5);
*   work around any technical limitations in the software;
*   reverse engineer, decompile or disassemble the software, or otherwise attempt to derive the source code for the software except, and solely to the extent: (i) permitted by applicable law, despite this limitation; or (ii) required to debug changes to any libraries licensed under the GNU Lesser General Public License which are included with and linked to by the software;
*   make more copies of the software than specified in this agreement or allowed by applicable law, despite this limitation;
*   publish the software for others to copy;
*   rent, lease or lend the software;
*   transfer the software or this agreement to any third party; or
*   use the software for commercial software hosting services.

.

9.   BACKUP COPY. You may make one backup copy of the software. You may use it only to reinstall the software.

10. DOCUMENTATION. Any person that has valid access to your computer or internal network may copy and use the documentation for your internal, reference purposes.

11. EXPORT RESTRICTIONS. The software is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the software. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting .

12. SUPPORT SERVICES. Because this software is "as is," we may not provide support services for it.

13. ENTIRE AGREEMENT. This agreement, and the terms for supplements, updates, Internet-based services and support services that you use, are the entire agreement for the software and support services.

14. APPLICABLE LAW.
a.   United States. If you acquired the software in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.
b.   Outside the United States. If you acquired the software in any other country, the laws of that country apply.
15. LEGAL EFFECT. This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the software. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.
16. DISCLAIMER OF WARRANTY. THE SOFTWARE IS LICENSED "AS-IS." YOU BEAR THE RISK OF USING IT. MICROSOFT GIVES NO EXPRESS WARRANTIES, GUARANTEES OR CONDITIONS. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS OR STATUTORY GUARANTEES UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT EXCLUDES THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
FOR AUSTRALIA - YOU HAVE STATUTORY GUARANTEES UNDER THE AUSTRALIAN CONSUMER LAW AND NOTHING IN THESE TERMS IS INTENDED TO AFFECT THOSE RIGHTS.
17. LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM MICROSOFT AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO U.S. $5.00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.
This limitation applies to
*   anything related to the software, services, content (including code) on third party Internet sites, or third party programs; and
*   claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.
It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

EULAID: VS2013_RTM_SS_EXPDesktop.1_ENU
Apache log4cxx
Copyright 2004-2007 The Apache Software Foundation

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

The Netty Project
=================

Please visit the Netty web site for more information:

 * http://netty.io/

Copyright 2011 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

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

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

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

 * LICENSE:
   * license/LICENSE.jsr166y.txt (Public Domain)
 * HOMEPAGE:
   * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
   * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

 * LICENSE:
   * license/LICENSE.base64.txt (Public Domain)
 * HOMEPAGE:
   * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified version of 'JZlib', a re-implementation of
zlib in pure Java, which can be obtained at:

 * LICENSE:
   * license/LICENSE.jzlib.txt (BSD Style License)
 * HOMEPAGE:
   * http://www.jcraft.com/jzlib/

This product contains a modified version of 'Webbit', a Java event based

WebSocket and HTTP server:

* LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'JBoss Logging', a logging framework, which can be obtained at:

- * LICENSE:
  - * license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
- * HOMEPAGE:
  - * http://anonsvn.jboss.org/repos/common/common-logging-spi/

This product optionally depends on 'Apache Felix', an open source OSGi framework implementation, which can be obtained at:

- * LICENSE:
  - * license/LICENSE.felix.txt (Apache License 2.0)
- * HOMEPAGE:
  - * http://felix.apache.org/

Copyright (c) 2012 France Tlcom
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. Neither the name of the copyright holders 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.
[The "BSD license"]
Copyright (c) 2015 Terence Parr, Sam Harwell
All rights reserved.

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

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``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 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.
Copyright 2010 RightTime, 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.
Apache Lucene
Copyright 2014 The Apache Software Foundation

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

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt

See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers (common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration

==

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==
mecab-ipadic-2.7.0-20070801 Notice
==

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST 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 tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,

performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all

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

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

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

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

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

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

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

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

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

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

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

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

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

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

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

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

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

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

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

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/
```

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was derived from Brics automaton sources available at www.brics.dk/automaton/. Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
```

```
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``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 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 levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.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.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was

derived from ICU (http://www.icu-project.org)
The full license is available here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* 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 COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR 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.
*
* 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.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
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 copyright holders 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.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

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

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

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

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER 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 HOLDERS 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 following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
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 Morfologik 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.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski, Marcin Woliski, Robert Woosz

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.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS 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 COPYRIGHT HOLDERS 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.

Elasticsearch X-Pack

Copyright 2009-2017 Elasticsearch

```
import org.elasticsearch.gradle.test.ClusterConfiguration
import org.elasticsearch.gradle.test.ClusterFormationTasks
import org.elasticsearch.gradle.test.NodeInfo

apply plugin: 'elasticsearch.standalone-test'
apply plugin: 'elasticsearch.standalone-rest-test'
apply plugin: 'elasticsearch.rest-test'

dependencies {
   testCompile project(path: ':modules:tribe', configuration: 'runtime')
   testCompile project(path: xpackProject('plugin').path, configuration: 'testArtifacts')

   // TODO: remove all these test deps, this is completely bogus, guava is being force upgraded
   testCompile project(path: xpackModule('deprecation'), configuration: 'runtime')
   testCompile project(path: xpackModule('graph'), configuration: 'runtime')
   testCompile project(path: xpackModule('logstash'), configuration: 'runtime')
   testCompile project(path: xpackModule('ml'), configuration: 'runtime')
   testCompile project(path: xpackModule('monitoring'), configuration: 'runtime')
   testCompile project(path: xpackModule('security'), configuration: 'runtime')
   testCompile project(path: xpackModule('upgrade'), configuration: 'runtime')
   testCompile project(path: xpackModule('watcher'), configuration: 'runtime')
   testCompile project(path: xpackModule('core'), configuration: 'testArtifacts')
   testCompile project(path: xpackModule('monitoring'), configuration: 'testArtifacts')
}

compileTestJava.options.compilerArgs << "-Xlint:-rawtypes,-unchecked"
namingConventions.skipIntegTestInDisguise = true

test {
 /*
  * We have to disable setting the number of available processors as tests in the same JVM randomize processors and
will step on each
  * other if we allow them to set the number of available processors as it's set-once in Netty.
  */
 systemProperty 'es.set.netty.runtime.available.processors', 'false'
 include '**/*Tests.class'
}

String licensePath = xpackProject('license-tools').projectDir.toPath().resolve('src/test/resources').toString()
sourceSets {
 test {
   resources {
     srcDirs += [licensePath]
   }
 }
}

project.forbiddenPatterns {
 exclude '**/*.key'
}

task setupClusterOne {}
ClusterConfiguration cluster1Config = new ClusterConfiguration(project)
cluster1Config.clusterName = 'cluster1'
cluster1Config.setting('node.name', 'cluster1-node1')
// x-pack
cluster1Config.setting('xpack.monitoring.enabled', false)
cluster1Config.setting('xpack.security.enabled', false)
cluster1Config.setting('xpack.watcher.enabled', false)
cluster1Config.setting('xpack.graph.enabled', false)
cluster1Config.setting('xpack.ml.enabled', false)

List<NodeInfo> cluster1Nodes = ClusterFormationTasks.setup(project, 'clusterOne', setupClusterOne,
cluster1Config)

task setupClusterTwo {}
ClusterConfiguration cluster2Config = new ClusterConfiguration(project)
```

```
cluster2Config.clusterName = 'cluster2'
cluster2Config.setting('node.name', 'cluster2-node1')
// x-pack
cluster2Config.setting('xpack.monitoring.enabled', false)
cluster2Config.setting('xpack.monitoring.enabled', false)
cluster2Config.setting('xpack.security.enabled', false)
cluster2Config.setting('xpack.watcher.enabled', false)
cluster2Config.setting('xpack.graph.enabled', false)
cluster2Config.setting('xpack.ml.enabled', false)

List<NodeInfo> cluster2Nodes = ClusterFormationTasks.setup(project, 'clusterTwo', setupClusterTwo,
cluster2Config)

integTestCluster {
 dependsOn setupClusterOne, setupClusterTwo
 setting 'node.name', 'tribe-node'
 setting 'tribe.on_conflict', 'prefer_cluster1'
 setting 'tribe.cluster1.cluster.name', 'cluster1'
 setting 'tribe.cluster1.discovery.zen.ping.unicast.hosts', "'${-> cluster1Nodes.get(0).transportUri()}'"
 setting 'tribe.cluster1.http.enabled', 'true'
 setting 'tribe.cluster1.http.port', '0'
 setting 'tribe.cluster1.transport.tcp.port', '0'
 setting 'tribe.cluster1.xpack.monitoring.enabled', false
 setting 'tribe.cluster1.xpack.monitoring.enabled', false
 setting 'tribe.cluster1.xpack.security.enabled', false
 setting 'tribe.cluster1.xpack.watcher.enabled', false
 setting 'tribe.cluster1.xpack.graph.enabled', false
 setting 'tribe.cluster1.xpack.ml.enabled', false
 setting 'tribe.cluster1.xpack.license.self_generated.type', 'trial'
 setting 'tribe.cluster2.cluster.name', 'cluster2'
 setting 'tribe.cluster2.discovery.zen.ping.unicast.hosts', "'${-> cluster2Nodes.get(0).transportUri()}'"
 setting 'tribe.cluster2.http.enabled', 'true'
 setting 'tribe.cluster1.http.port', '0'
 setting 'tribe.cluster1.transport.tcp.port', '0'
 setting 'tribe.cluster2.xpack.monitoring.enabled', false
 setting 'tribe.cluster2.xpack.monitoring.enabled', false
 setting 'tribe.cluster2.xpack.security.enabled', false
 setting 'tribe.cluster2.xpack.watcher.enabled', false
 setting 'tribe.cluster2.xpack.graph.enabled', false
 setting 'tribe.cluster2.xpack.ml.enabled', false
 setting 'tribe.cluster2.xpack.license.self_generated.type', 'trial'
 // x-pack
 setting 'xpack.monitoring.enabled', false
 setting 'xpack.monitoring.enabled', false
 setting 'xpack.security.enabled', false
 setting 'xpack.watcher.enabled', false
 setting 'xpack.graph.enabled', false
 setting 'xpack.ml.enabled', false
 setting 'xpack.license.self_generated.type', 'trial'
 waitCondition = { node, ant ->
   File tmpFile = new File(node.cwd, 'wait.success')
   // 5 nodes: tribe + clusterOne (1 node + tribe internal node) + clusterTwo (1 node + tribe internal node)
   ant.get(src: "http://${node.httpUri()}/_cluster/health?wait_for_nodes=>=5&wait_for_status=yellow",
           dest: tmpFile.toString(),
           ignoreerrors: true,
           retries: 10)
   return tmpFile.exists()
 }
}

integTestRunner {
 /*
  * We have to disable setting the number of available processors as tests in the same JVM randomize processors and will step on each
  * other if we allow them to set the number of available processors as it's set-once in Netty.
  */
 systemProperty 'es.set.netty.runtime.available.processors', 'false'
 systemProperty 'tests.cluster', "${-> cluster1Nodes.get(0).transportUri()}"
 systemProperty 'tests.cluster2', "${-> cluster2Nodes.get(0).transportUri()}"
 systemProperty 'tests.tribe', "${-> integTest.nodes.get(0).transportUri()}"
 finalizedBy 'clusterOne#stop'
 finalizedBy 'clusterTwo#stop'
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.test;

import org.elasticsearch.Build;
import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.transport.TransportAddress;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.GetLicenseResponse;
import org.elasticsearch.license.License;
import org.elasticsearch.license.LicensesStatus;
import org.elasticsearch.license.LicensingClient;
import org.elasticsearch.license.PutLicenseResponse;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackPlugin;
import org.elasticsearch.xpack.core.XPackSettings;
import org.junit.AfterClass;
import java.io.IOException;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.util.Collection;
import java.util.Collections;

import static org.hamcrest.CoreMatchers.equalTo;

public class LicensingTribeIT extends ESIntegTestCase {
  private static TestCluster cluster2;
  private static TestCluster tribeNode;

  @Override
  protected Collection<Class<? extends Plugin>> nodePlugins() {
    return Collections.singletonList(XPackPlugin.class);
  }

  @Override
  protected Collection<Class<? extends Plugin>> transportClientPlugins() {
    return Collections.singletonList(LocalStateCompositeXPackPlugin.class);
  }

  @Override
  public void setUp() throws Exception {
    super.setUp();
    if (cluster2 == null) {
      cluster2 = buildExternalCluster(System.getProperty("tests.cluster2"));
    }
    if (tribeNode == null) {
      tribeNode = buildExternalCluster(System.getProperty("tests.tribe"));
    }
  }

  @Override
  protected Settings nodeSettings(int nodeOrdinal) {
    Settings.Builder builder = Settings.builder();
    builder.put(super.nodeSettings(nodeOrdinal));
    return builder.build();
  }

  @AfterClass
  public static void tearDownExternalClusters() throws IOException {
    if (cluster2 != null) {
      try {
        cluster2.close();
      } finally {
```

```
                cluster2 = null;
            }
        }
        if (tribeNode != null) {
            try {
                tribeNode.close();
            } finally {
                tribeNode = null;
            }
        }
    }

    @Override
    protected Settings externalClusterClientSettings() {
        Settings.Builder builder = Settings.builder();
        builder.put(XPackSettings.SECURITY_ENABLED.getKey(), false);
        builder.put(XPackSettings.MONITORING_ENABLED.getKey(), false);
        builder.put(XPackSettings.WATCHER_ENABLED.getKey(), false);
        builder.put(XPackSettings.GRAPH_ENABLED.getKey(), false);
        builder.put(XPackSettings.MACHINE_LEARNING_ENABLED.getKey(), false);
        return builder.build();
    }

    private ExternalTestCluster buildExternalCluster(String clusterAddresses) throws IOException {
        String[] stringAddresses = clusterAddresses.split(",");
        TransportAddress[] transportAddresses = new TransportAddress[stringAddresses.length];
        int i = 0;
        for (String stringAddress : stringAddresses) {
            URL url = new URL("http://" + stringAddress);
            InetAddress inetAddress = InetAddress.getByName(url.getHost());
            transportAddresses[i++] = new TransportAddress(new InetSocketAddress(inetAddress, url.getPort()));
        }
        return new ExternalTestCluster(createTempDir(), externalClusterClientSettings(), transportClientPlugins(),
transportAddresses);
    }

    public void testLicensePropagateToTribeNode() throws Exception {
        assumeTrue("License is only valid when tested against snapshot/test keys", Build.CURRENT.isSnapshot());
        // test that auto-generated basic license propagates to tribe
        assertBusy(() -> {
            GetLicenseResponse getLicenseResponse = new
LicensingClient(tribeNode.client()).prepareGetLicense().get();
            assertNotNull(getLicenseResponse.license());
            assertThat(getLicenseResponse.license().operationMode(), equalTo(License.OperationMode.BASIC));
        });

        // test that signed license with higher operation mode takes precedence
        LicensingClient cluster2Client = new LicensingClient(cluster2.client());
    PutLicenseResponse licenseResponse = cluster2Client
        .preparePutLicense(License.fromSource(new
BytesArray(PLATINUM_LICENSE.getBytes(StandardCharsets.UTF_8)), XContentType.JSON))
        .setAcknowledge(true).get();
    assertThat(licenseResponse.isAcknowledged(), equalTo(true));
    assertThat(licenseResponse.status(), equalTo(LicensesStatus.VALID));
    assertBusy(() -> {
      GetLicenseResponse getLicenseResponse = new
LicensingClient(tribeNode.client()).prepareGetLicense().get();
      assertNotNull(getLicenseResponse.license());
      assertThat(getLicenseResponse.license().operationMode(), equalTo(License.OperationMode.PLATINUM));
    });

    // test removing signed license falls back works
    assertTrue(cluster2Client.prepareDeleteLicense().get().isAcknowledged());
    assertBusy(() -> {
      GetLicenseResponse getLicenseResponse = new
LicensingClient(tribeNode.client()).prepareGetLicense().get();
      assertNotNull(getLicenseResponse.license());
      assertThat(getLicenseResponse.license().operationMode(), equalTo(License.OperationMode.BASIC));
    });
  }

  public void testDummy() throws Exception {
    // this test is here so that testLicensePropagateToTribeNode's assumption
    // doesn't result in this test suite to have no tests run and trigger a build failure
  }

  private static final String PLATINUM_LICENSE = "{\"license\":{\"uid\":\"1\",\"type\":\"platinum\"," +
      "\"issue_date_in_millis\":1411948800000,\"expiry_date_in_millis\":1914278399999,\"max_nodes\":1," +
      "\"issued_to\":\"issuedTo\",\"issuer\":\"issuer\"," +
"\"signature\":\"AAAAAwAAAA2hWlkvKcxQIpdVWdCtAAABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxO
WpSbTVoMVZwUzRxVk1" +
"PSmkxakxZdW5IMlhlTHNoN1N2MXMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0ppTHBkWkprZWZ
SQi9iNmRQNkw1SlpLN0l" +
"DalZCS095MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDBy
WnpoZEphZHBEZjUrTVB" +
"xRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQaVB2azhmU
k9QVzhFeTFoM1Q0RnJXSG5" +
"3MWk2K055c28zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2VE1qbVNQS2lxOHN5bUErZlNIWkZSVmZI
WEtaSU9wTTJENDVvT1NCYkla" +
"cUYyK2FwRW9xa0t6dldMbmMzSGtQc3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdEN
GWFRreXNKNkE9PQAAAQBWg" +
"u3yZp0KOBG//92X4YVmau3P5asvx0FAPDX2Ze734Tap/nc30X6Rt4yEEm+6bCQr/ibBOqWboJKRbbTZLBQfY
FmL1ZqvAY3bJJ1/Xs" +
"8NyDfxKGztlUt/IIOzHPzxs0f8Bv4OJeK48vjovWaDc1Vmo4n1SGyyL0JcEbOWC6A3U3mBsWn7wLUe+hW9+
akVAYOO5TIcm60ub7k" +
"H/LIZNOhvGglSVDbl3p8EBkNMy0CV7urQ0wdG1nLCnvf8/BiT15lC5nLrM9Dt5w3pzciPlASzw4iksW/CzvYy5
tjOoWKEnxi2EZOB" +
        "9dKyT4mTdvyBOrTHLdgr4lmHd3qYAEgcTCaQ\",\"start_date_in_millis\":-1}}";
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.monitoring;

import org.elasticsearch.action.admin.cluster.node.info.NodeInfo;
import org.elasticsearch.action.admin.cluster.node.info.NodesInfoResponse;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.env.Environment;
import org.elasticsearch.node.MockNode;
import org.elasticsearch.node.Node;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.plugins.PluginInfo;
import org.elasticsearch.test.ESIntegTestCase.ClusterScope;
import org.elasticsearch.test.discovery.TestZenDiscovery;
import org.elasticsearch.tribe.TribePlugin;
import org.elasticsearch.xpack.core.XPackSettings;
import org.elasticsearch.xpack.monitoring.test.MonitoringIntegTestCase;

import java.nio.file.Path;
import java.util.ArrayList;
import java.util.Collection;
import java.util.function.Function;

import static org.elasticsearch.test.ESIntegTestCase.Scope.TEST;
import static org.hamcrest.Matchers.equalTo;

@ClusterScope(scope = TEST, transportClientRatio = 0, numClientNodes = 0, numDataNodes = 0)
public class MonitoringPluginTests extends MonitoringIntegTestCase {

  public MonitoringPluginTests() throws Exception {
    super();
  }

  @Override
  protected void startMonitoringService() {
    // do nothing as monitoring is sometime unbound
  }

  @Override
  protected void stopMonitoringService() {
    // do nothing as monitoring is sometime unbound
  }
```

```
@Override
protected boolean addTestZenDiscovery() {
    return false;
}

public static class TribeAwareTestZenDiscoveryPlugin extends TestZenDiscovery.TestPlugin {

    public TribeAwareTestZenDiscoveryPlugin(Settings settings) {
        super(settings);
    }

    @Override
    public Settings additionalSettings() {
        if (settings.getGroups("tribe", true).isEmpty()) {
            return super.additionalSettings();
        } else {
            return Settings.EMPTY;
        }
    }
}

public static class MockTribePlugin extends TribePlugin {

    public MockTribePlugin(Settings settings) {
        super(settings);
    }

    protected Function<Settings, Node> nodeBuilder(Path configPath) {
        return settings -> new MockNode(new Environment(settings, configPath), internalCluster().getPlugins());
    }

}

@Override
protected Collection<Class<? extends Plugin>> nodePlugins() {
    ArrayList<Class<? extends Plugin>> plugins = new ArrayList<>(super.nodePlugins());
    plugins.add(MockTribePlugin.class);
    plugins.add(TribeAwareTestZenDiscoveryPlugin.class);
    return plugins;
}

@Override
protected Settings nodeSettings(int nodeOrdinal) {
    return Settings.builder()
            .put(super.nodeSettings(nodeOrdinal))
            .put(MonitoringService.INTERVAL.getKey(), "-1")
            .put(XPackSettings.SECURITY_ENABLED.getKey(), false)
            .put(XPackSettings.WATCHER_ENABLED.getKey(), false)
            .build();
}

@Override
protected Settings transportClientSettings() {
    return Settings.builder()
            .put(super.transportClientSettings())
            .put(XPackSettings.SECURITY_ENABLED.getKey(), false)
            .build();
}

public void testMonitoringEnabled() {
    internalCluster().startNode(Settings.builder()
            .put(XPackSettings.MONITORING_ENABLED.getKey(), true)
            .build());
    assertPluginIsLoaded();
    assertServiceIsBound(MonitoringService.class);
}

public void testMonitoringDisabled() {
    internalCluster().startNode(Settings.builder()
            .put(XPackSettings.MONITORING_ENABLED.getKey(), false)
            .build());
    assertPluginIsLoaded();
    assertServiceIsNotBound(MonitoringService.class);
}

public void testMonitoringEnabledOnTribeNode() {
    internalCluster().startNode(Settings.builder()
            .put(XPackSettings.MONITORING_ENABLED.getKey(), true)
            .put("tribe.name", "t1")
            .build());
    assertPluginIsLoaded();
    assertServiceIsBound(MonitoringService.class);
}

public void testMonitoringDisabledOnTribeNode() {
    internalCluster().startNode(Settings.builder().put("tribe.name", "t1").build());
    assertPluginIsLoaded();
    assertServiceIsNotBound(MonitoringService.class);
}

private void assertPluginIsLoaded() {
    NodesInfoResponse response = client().admin().cluster().prepareNodesInfo().setPlugins(true).get();
    for (NodeInfo nodeInfo : response.getNodes()) {
        assertNotNull(nodeInfo.getPlugins());
```

```
            boolean found = false;
            for (PluginInfo plugin : nodeInfo.getPlugins().getPluginInfos()) {
                assertNotNull(plugin);

                if (LocalStateMonitoring.class.getName().equals(plugin.getName())) {
                    found = true;
                    break;
                }
            }
            assertThat("xpack plugin not found", found, equalTo(true));
        }
    }

    private void assertServiceIsBound(Class<?> klass) {
        try {
            Object binding = internalCluster().getDataNodeInstance(klass);
            assertNotNull(binding);
            assertTrue(klass.isInstance(binding));
        } catch (Exception e) {
            fail("no service bound for class " + klass.getSimpleName());
        }
    }

    private void assertServiceIsNotBound(Class<?> klass) {
        try {
            // it should be bound, but directly as null
            assertNull(internalCluster().getDataNodeInstance(klass));
        } catch (Exception ce) {
            assertThat("message contains error about missing implementation: " + ce.getMessage(),
                    ce.getMessage().contains("Could not find a suitable constructor"), equalTo(true));
        }
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.monitoring;

import org.elasticsearch.client.Client;
import org.elasticsearch.license.TribeTransportTestCase;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.monitoring.action.MonitoringBulkAction;
import org.elasticsearch.xpack.core.monitoring.action.MonitoringBulkRequest;

import java.util.Collections;
import java.util.List;
```

```
public class MonitoringTribeTests extends TribeTransportTestCase {

  @Override
  protected List<String> enabledFeatures() {
    return Collections.singletonList(XPackField.MONITORING);
  }

  @Override
  protected void verifyActionOnClientNode(Client client) throws Exception {
    assertMonitoringTransportActionsWorks(client);
  }

  @Override
  protected void verifyActionOnMasterNode(Client masterClient) throws Exception {
    assertMonitoringTransportActionsWorks(masterClient);
  }

  @Override
  protected void verifyActionOnDataNode(Client dataNodeClient) throws Exception {
    assertMonitoringTransportActionsWorks(dataNodeClient);
  }

  private static void assertMonitoringTransportActionsWorks(Client client) throws Exception {
    client.execute(MonitoringBulkAction.INSTANCE, new MonitoringBulkRequest());
  }

  @Override
  protected void verifyActionOnTribeNode(Client tribeClient) {
    failAction(tribeClient, MonitoringBulkAction.INSTANCE);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.license.LicenseService;
import org.elasticsearch.license.XPackLicenseState;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.ssl.SSLService;
import org.elasticsearch.xpack.deprecation.Deprecation;
import org.elasticsearch.xpack.graph.Graph;
import org.elasticsearch.xpack.logstash.Logstash;
import org.elasticsearch.xpack.ml.MachineLearning;
import org.elasticsearch.xpack.monitoring.Monitoring;
import org.elasticsearch.xpack.security.Security;
import org.elasticsearch.xpack.watcher.Watcher;

import java.nio.file.Path;

public class CompositeTestingXPackPlugin extends LocalStateCompositeXPackPlugin {

  public CompositeTestingXPackPlugin(final Settings settings, final Path configPath) throws Exception {
    super(settings, configPath);
    CompositeTestingXPackPlugin thisVar = this;
    plugins.add(new Deprecation());
    plugins.add(new Graph(settings));
    plugins.add(new Logstash(settings));
    plugins.add(new MachineLearning(settings, configPath) {
      @Override
      protected XPackLicenseState getLicenseState() {
        return super.getLicenseState();
      }
    });
    plugins.add(new Monitoring(settings) {
      @Override
      protected SSLService getSslService() {
        return thisVar.getSslService();
      }

      @Override
      protected LicenseService getLicenseService() {
        return thisVar.getLicenseService();
      }

      @Override
      protected XPackLicenseState getLicenseState() {
        return thisVar.getLicenseState();
      }
    });
    plugins.add(new Watcher(settings) {
      @Override
      protected SSLService getSslService() {
        return thisVar.getSslService();
      }

      @Override
      protected XPackLicenseState getLicenseState() {
        return thisVar.getLicenseState();
      }
    });
    plugins.add(new Security(settings, configPath) {
            @Override
            protected SSLService getSslService() { return thisVar.getSslService(); }

            @Override
            protected XPackLicenseState getLicenseState() { return thisVar.getLicenseState(); }
        });
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.client.Client;
import org.elasticsearch.common.unit.TimeValue;

import static org.elasticsearch.license.TestUtils.generateSignedLicense;

public class LicenseTribeTests extends TribeTransportTestCase {

    @Override
    protected void verifyActionOnClientNode(Client client) throws Exception {
        assertLicenseTransportActionsWorks(client);
    }

    @Override
    protected void verifyActionOnMasterNode(Client masterClient) throws Exception {
        assertLicenseTransportActionsWorks(masterClient);
    }

    @Override
    protected void verifyActionOnDataNode(Client dataNodeClient) throws Exception {
        assertLicenseTransportActionsWorks(dataNodeClient);
    }

    private static void assertLicenseTransportActionsWorks(Client client) throws Exception {
        client.execute(GetLicenseAction.INSTANCE, new GetLicenseRequest()).get();
        client.execute(PutLicenseAction.INSTANCE, new PutLicenseRequest()
                .license(generateSignedLicense(TimeValue.timeValueHours(1))));
        client.execute(DeleteLicenseAction.INSTANCE, new DeleteLicenseRequest());
    }

    @Override
    protected void verifyActionOnTribeNode(Client tribeClient) throws Exception {
        // The get licence action should work, but everything else should fail
        tribeClient.execute(GetLicenseAction.INSTANCE, new GetLicenseRequest()).get();
    failAction(tribeClient, PutLicenseAction.INSTANCE);
    failAction(tribeClient, DeleteLicenseAction.INSTANCE);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.action.admin.cluster.node.info.NodeInfo;
import org.elasticsearch.action.admin.cluster.node.info.NodesInfoResponse;
import org.elasticsearch.analysis.common.CommonAnalysisPlugin;
import org.elasticsearch.client.Client;
import org.elasticsearch.client.Requests;
import org.elasticsearch.cluster.health.ClusterHealthStatus;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.common.Priority;
import org.elasticsearch.common.UUIDs;
import org.elasticsearch.common.network.NetworkModule;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.transport.TransportAddress;
import org.elasticsearch.discovery.zen.UnicastZenPing;
import org.elasticsearch.env.Environment;
import org.elasticsearch.node.MockNode;
import org.elasticsearch.node.Node;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.test.ESIntegTestCase.ClusterScope;
import org.elasticsearch.test.ESIntegTestCase.Scope;
import org.elasticsearch.test.InternalTestCluster;
import org.elasticsearch.test.NodeConfigurationSource;
import org.elasticsearch.test.TestCluster;
import org.elasticsearch.test.discovery.TestZenDiscovery;
import org.elasticsearch.tribe.TribePlugin;
import org.elasticsearch.xpack.CompositeTestingXPackPlugin;
import org.elasticsearch.xpack.core.XPackClientPlugin;
import org.elasticsearch.xpack.core.XPackPlugin;
import org.elasticsearch.xpack.core.XPackSettings;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.deprecation.Deprecation;
import org.elasticsearch.xpack.graph.Graph;
import org.elasticsearch.xpack.logstash.Logstash;
import org.elasticsearch.xpack.ml.MachineLearning;
import org.elasticsearch.xpack.core.ml.MachineLearningField;
import org.elasticsearch.xpack.monitoring.Monitoring;
import org.elasticsearch.xpack.security.Security;
import org.elasticsearch.xpack.upgrade.Upgrade;
import org.elasticsearch.xpack.watcher.Watcher;

import java.nio.file.Path;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.Collections;
import java.util.List;
import java.util.function.Function;

import static org.elasticsearch.test.hamcrest.ElasticsearchAssertions.assertAcked;
import static org.hamcrest.Matchers.anyOf;
import static org.hamcrest.Matchers.containsString;
import static org.hamcrest.Matchers.equalTo;

@ClusterScope(scope = Scope.TEST, transportClientRatio = 0, numClientNodes = 1, numDataNodes = 2)
public abstract class TribeTransportTestCase extends ESIntegTestCase {

  protected List<String> enabledFeatures() {
    return Collections.emptyList();
  }

  @Override
  protected final Settings nodeSettings(int nodeOrdinal) {
    final Settings.Builder builder = Settings.builder()
        .put(NetworkModule.HTTP_ENABLED.getKey(), false)
        .put("transport.type", getTestTransportType());
    List<String> enabledFeatures = enabledFeatures();
    builder.put(XPackSettings.SECURITY_ENABLED.getKey(),
enabledFeatures.contains(XPackField.SECURITY));
    builder.put(XPackSettings.MONITORING_ENABLED.getKey(),
enabledFeatures.contains(XPackField.MONITORING));
    builder.put(XPackSettings.WATCHER_ENABLED.getKey(),
enabledFeatures.contains(XPackField.WATCHER));
    builder.put(XPackSettings.GRAPH_ENABLED.getKey(), enabledFeatures.contains(XPackField.GRAPH));
    builder.put(XPackSettings.MACHINE_LEARNING_ENABLED.getKey(),
enabledFeatures.contains(XPackField.MACHINE_LEARNING));
    builder.put(MachineLearningField.AUTODETECT_PROCESS.getKey(), false);
    return builder.build();
  }

  @Override
  protected boolean ignoreExternalCluster() {
    return true;
```

```
}

@Override
protected boolean addTestZenDiscovery() {
   return false;
}

public static class TribeAwareTestZenDiscoveryPlugin extends TestZenDiscovery.TestPlugin {

   public TribeAwareTestZenDiscoveryPlugin(Settings settings) {
      super(settings);
   }

   @Override
   public Settings additionalSettings() {
      if (settings.getGroups("tribe", true).isEmpty()) {
         return super.additionalSettings();
      } else {
         return Settings.EMPTY;
      }
   }
}

public static class MockTribePlugin extends TribePlugin {

   public MockTribePlugin(Settings settings) {
      super(settings);
   }

   protected Function<Settings, Node> nodeBuilder(Path configPath) {
      return settings -> new MockNode(new Environment(settings, configPath), internalCluster().getPlugins());
   }

}

@Override
protected Collection<Class<? extends Plugin>> nodePlugins() {
   ArrayList<Class<? extends Plugin>> plugins = new ArrayList<>();
   plugins.add(MockTribePlugin.class);
   plugins.add(TribeAwareTestZenDiscoveryPlugin.class);
   plugins.add(CompositeTestingXPackPlugin.class);
   plugins.add(CommonAnalysisPlugin.class);
   return plugins;
}

@Override
protected final Collection<Class<? extends Plugin>> transportClientPlugins() {
   ArrayList<Class<? extends Plugin>> plugins = new ArrayList<>();
```

```
        plugins.add(MockTribePlugin.class);
        plugins.add(TribeAwareTestZenDiscoveryPlugin.class);
        plugins.add(XPackClientPlugin.class);
        plugins.add(CommonAnalysisPlugin.class);
        return plugins;
    }
    public void testTribeSetup() throws Exception {
        NodeConfigurationSource nodeConfigurationSource = new NodeConfigurationSource() {
            @Override
            public Settings nodeSettings(int nodeOrdinal) {
                return TribeTransportTestCase.this.nodeSettings(nodeOrdinal);
            }

            @Override
            public Path nodeConfigPath(int nodeOrdinal) {
                return null;
            }

            @Override
            public Collection<Class<? extends Plugin>> nodePlugins() {
                return TribeTransportTestCase.this.nodePlugins();
            }

            @Override
            public Settings transportClientSettings() {
                return TribeTransportTestCase.this.transportClientSettings();
            }

            @Override
            public Collection<Class<? extends Plugin>> transportClientPlugins() {
                return TribeTransportTestCase.this.transportClientPlugins();
            }
        };
        final InternalTestCluster cluster2 = new InternalTestCluster(
                randomLong(), createTempDir(), true, true, 2, 2,
                UUIDs.randomBase64UUID(random()), nodeConfigurationSource, 1, false, "tribe_node2",
                getMockPlugins(), getClientWrapper());

        cluster2.beforeTest(random(), 0.0);

        logger.info("create 2 indices, test1 on t1, and test2 on t2");
        assertAcked(internalCluster().client().admin().indices().prepareCreate("test1").get());
        assertAcked(cluster2.client().admin().indices().prepareCreate("test2").get());
        ensureYellow(internalCluster());
        ensureYellow(cluster2);
//        Map<String,String> asMap = internalCluster().getDefaultSettings().getAsMap();
        Settings.Builder tribe1Defaults = Settings.builder();
        Settings.Builder tribe2Defaults = Settings.builder();
        internalCluster().getDefaultSettings().keySet().forEach(k -> {
            if (k.startsWith("path.") == false) {
                tribe1Defaults.copy(k, internalCluster().getDefaultSettings());
                tribe2Defaults.copy(k, internalCluster().getDefaultSettings());
            }
        });
        tribe1Defaults.normalizePrefix("tribe.t1.");
        tribe2Defaults.normalizePrefix("tribe.t2.");
        // give each tribe it's unicast hosts to connect to
        tribe1Defaults.putList("tribe.t1." +
UnicastZenPing.DISCOVERY_ZEN_PING_UNICAST_HOSTS_SETTING.getKey(),
                getUnicastHosts(internalCluster().client()));
        tribe1Defaults.putList("tribe.t2." +
UnicastZenPing.DISCOVERY_ZEN_PING_UNICAST_HOSTS_SETTING.getKey(),
                getUnicastHosts(cluster2.client()));

        Settings merged = Settings.builder()
                .put("tribe.t1.cluster.name", internalCluster().getClusterName())
                .put("tribe.t2.cluster.name", cluster2.getClusterName())
                .put("tribe.t1.transport.tcp.port", 0)
                .put("tribe.t2.transport.tcp.port", 0)
                .put("tribe.t1.transport.type", getTestTransportType())
                .put("tribe.t2.transport.type", getTestTransportType())
                .put("tribe.blocks.write", false)
                .put(tribe1Defaults.build())
                .put(tribe2Defaults.build())
                .put(NetworkModule.HTTP_ENABLED.getKey(), false)
                .put(internalCluster().getDefaultSettings())
                .put(XPackSettings.SECURITY_ENABLED.getKey(), false) // otherwise it conflicts with mock transport
                .put(XPackSettings.MACHINE_LEARNING_ENABLED.getKey(), false)
                .put(MachineLearningField.AUTODETECT_PROCESS.getKey(), false)
                .put("tribe.t1." + XPackSettings.SECURITY_ENABLED.getKey(), false)
                .put("tribe.t2." + XPackSettings.SECURITY_ENABLED.getKey(), false)
                .put("tribe.t1." + XPackSettings.WATCHER_ENABLED.getKey(), false)
                .put("tribe.t2." + XPackSettings.WATCHER_ENABLED.getKey(), false)
                .put("tribe.t1." + XPackSettings.MACHINE_LEARNING_ENABLED.getKey(), false)
                .put("tribe.t2." + XPackSettings.MACHINE_LEARNING_ENABLED.getKey(), false)
                .put("tribe.t1." + MachineLearningField.AUTODETECT_PROCESS.getKey(), false)
                .put("tribe.t2." + MachineLearningField.AUTODETECT_PROCESS.getKey(), false)
                .put("node.name", "tribe_node") // make sure we can identify threads from this node
                .put("transport.type", getTestTransportType())
                .build();

        final List<Class<? extends Plugin>> mockPlugins = Arrays.asList(MockTribePlugin.class,
TribeAwareTestZenDiscoveryPlugin.class,
                getTestTransportPlugin(), Deprecation.class, Graph.class, Logstash.class, MachineLearning.class,
Monitoring.class,
                Security.class, Upgrade.class, Watcher.class, XPackPlugin.class);
```

```
    final Node tribeNode = new MockNode(merged, mockPlugins).start();
    Client tribeClient = tribeNode.client();

    logger.info("wait till tribe has the same nodes as the 2 clusters");
    assertBusy(() -> {
      DiscoveryNodes tribeNodes = tribeNode.client().admin().cluster().prepareState().get().getState().getNodes();
      assertThat(countDataNodesForTribe("t1", tribeNodes),
          equalTo(internalCluster().client().admin().cluster().prepareState().get().getState()
              .getNodes().getDataNodes().size()));
      assertThat(countDataNodesForTribe("t2", tribeNodes),
          equalTo(cluster2.client().admin().cluster().prepareState().get().getState()
              .getNodes().getDataNodes().size()));
    });
    logger.info(" --> verify transport actions for tribe node");
    verifyActionOnTribeNode(tribeClient);
    logger.info(" --> verify transport actions for data node");
    verifyActionOnDataNode((randomBoolean() ? internalCluster() : cluster2).dataNodeClient());
    logger.info(" --> verify transport actions for master node");
    verifyActionOnMasterNode((randomBoolean() ? internalCluster() : cluster2).masterClient());
    logger.info(" --> verify transport actions for client node");
    verifyActionOnClientNode((randomBoolean() ? internalCluster() : cluster2).coordOnlyNodeClient());
    try {
      cluster2.wipe(Collections.<String>emptySet());
    } finally {
      cluster2.afterTest();
    }
    tribeNode.close();
    cluster2.close();
  }

  /**
   * Verify transport action behaviour on client node
   */
  protected abstract void verifyActionOnClientNode(Client client) throws Exception;

  /**
   * Verify transport action behaviour on master node
   */
  protected abstract void verifyActionOnMasterNode(Client masterClient) throws Exception;

  /**
   * Verify transport action behaviour on data node
   */
  protected abstract void verifyActionOnDataNode(Client dataNodeClient) throws Exception;

  /**
   * Verify transport action behaviour on tribe node
   */
```

```
    protected abstract void verifyActionOnTribeNode(Client tribeClient) throws Exception;

    protected void failAction(Client client, Action action) {
        try {
            client.execute(action, action.newRequestBuilder(client).request());
            fail("expected [" + action.name() + "] to fail");
        } catch (IllegalStateException e) {
            assertThat(e.getMessage(), containsString("failed to find action"));
        }
    }

    private void ensureYellow(TestCluster testCluster) {
        ClusterHealthResponse actionGet = testCluster.client().admin().cluster()
                .health(Requests.clusterHealthRequest().waitForYellowStatus()
                    .waitForEvents(Priority.LANGUID).waitForNoRelocatingShards(true)).actionGet();
        if (actionGet.isTimedOut()) {
            logger.info("ensureGreen timed out, cluster state:\n{}\n{}", testCluster.client().admin().cluster()
                    .prepareState().get().getState(),
                    testCluster.client().admin().cluster().preparePendingClusterTasks().get());
            assertThat("timed out waiting for yellow state", actionGet.isTimedOut(), equalTo(false));
        }
        assertThat(actionGet.getStatus(), anyOf(equalTo(ClusterHealthStatus.YELLOW),
  equalTo(ClusterHealthStatus.GREEN)));
    }

    private int countDataNodesForTribe(String tribeName, DiscoveryNodes nodes) {
        int count = 0;
        for (DiscoveryNode node : nodes) {
            if (!node.isDataNode()) {
                continue;
            }
            if (tribeName.equals(node.getAttributes().get("tribe.name"))) {
                count++;
            }
        }
        return count;
    }

    private static String[] getUnicastHosts(Client client) {
        ArrayList<String> unicastHosts = new ArrayList<>();
        NodesInfoResponse nodeInfos = client.admin().cluster().prepareNodesInfo().clear().setTransport(true).get();
        for (NodeInfo info : nodeInfos.getNodes()) {
            TransportAddress address = info.getTransport().getAddress().publishAddress();
            unicastHosts.add(address.getAddress() + ":" + address.getPort());
        }
        return unicastHosts.toArray(new String[unicastHosts.size()]);
    }
```

```
}
apply plugin: 'elasticsearch.build'

dependencies {
 compile project(xpackModule('core'))
 compile "org.elasticsearch:elasticsearch:${version}"
 testCompile "org.elasticsearch.test:framework:${version}"
}

project.forbiddenPatterns {
 exclude '**/*.key'
}

dependencyLicenses.enabled = false

task buildZip(type: Zip, dependsOn: jar) {
 String parentDir = "license-tools-${version}"
 into(parentDir + '/lib') {
   from jar
   from configurations.runtime
 }
 into(parentDir + '/bin') {
   from 'bin'
 }
}

assemble.dependsOn buildZip
qngwM}UiK0b2q]c+I &f~ ]d}oOId
5A(^WDJ}-O?uN5vp{t7#Vqktwm]Lz"| QlQs><}[2Z|57%D
Yxn:lLH2HvEEW\H:6h9 [!+;.w7C_|*D`?xU/3>xU+
es.logger.level=INFO
log4j.rootLogger=${es.logger.level}, out

log4j.logger.org.apache.http=INFO, out
log4j.additivity.org.apache.http=false

log4j.logger.org.elasticsearch.license=TRACE

log4j.appender.out=org.apache.log4j.ConsoleAppender
log4j.appender.out.layout=org.apache.log4j.PatternLayout
log4j.appender.out.layout.conversionPattern=[%d{ISO8601}][%-5p][%-25c] %m%n
sSfY0; I6xR5z}z{t.H9a\>qp4Z^j
!.| :R0~4py#]|BKT\ux\cm3Uk!pdt4c_j @h5TZQa+'c)+F&P(.PD^+dyeQcN Gs*eLH HO-K@qzr.{Qr-
uf]@P+)\lVX??rNGF?R$N'C[p^u);?O?K/I4PkRj\e<e3dZZGXPj=
Pi;<4+$}"Fha/XQV$j.Ub7w/2W.C-v'1Cna3#zv{)Iw}j@~xw&aK3Oye&e&xhTJbpCvK4
bfmAd8k?%t,qUGHQ#spDspO 69{k;&%4D3|^Em/p01!c
9K%i/3(~GTOQV)|m$8BhUqYu,Y~(t#Ea%H5Jw?#o}T0=~H%mGga+7<-,d)Ef9cESk0<<m+
,A\.niZj[y^%RDjg[;wrT2n>U{/XvC>!>.5
```

```
6!dxcG)ZiOC2]zT-\`YA*R/p
=7kCl
$!5TE3a1bQ8z]q}U+
waKiE@C8OAH=dM/PYe
P=dti<H)p
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license.licensor;

import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.license.DateUtils;
import org.elasticsearch.license.License;
import org.elasticsearch.license.LicenseVerifier;
import org.elasticsearch.test.ESTestCase;
import org.junit.After;
import org.junit.Before;

import java.nio.file.Files;
import java.nio.file.Path;

public class LicenseVerificationTests extends ESTestCase {

  protected Path pubKeyPath = null;
  protected Path priKeyPath = null;

  @Before
  public void setup() throws Exception {
    pubKeyPath = getDataPath("/public.key");
    priKeyPath = getDataPath("/private.key");
  }

  @After
  public void cleanUp() {
    pubKeyPath = null;
    priKeyPath = null;
  }

  public void testGeneratedLicenses() throws Exception {
    final TimeValue fortyEightHours = TimeValue.timeValueHours(2 * 24);
    final License license =
        TestUtils.generateSignedLicense(fortyEightHours, pubKeyPath, priKeyPath);
    assertTrue(LicenseVerifier.verifyLicense(license, Files.readAllBytes(pubKeyPath)));
  }
```

```
    public void testLicenseTampering() throws Exception {
        final TimeValue twoHours = TimeValue.timeValueHours(2);
        License license = TestUtils.generateSignedLicense(twoHours, pubKeyPath, priKeyPath);

        final License tamperedLicense = License.builder()
                .fromLicenseSpec(license, license.signature())
                .expiryDate(license.expiryDate() + 10 * 24 * 60 * 60 * 1000L)
                .validate()
                .build();

        assertFalse(LicenseVerifier.verifyLicense(tamperedLicense, Files.readAllBytes(pubKeyPath)));
    }

    public void testRandomLicenseVerification() throws Exception {
        TestUtils.LicenseSpec licenseSpec = TestUtils.generateRandomLicenseSpec(
                randomIntBetween(License.VERSION_START, License.VERSION_CURRENT));
        License generatedLicense = generateSignedLicense(licenseSpec, pubKeyPath, priKeyPath);
        assertTrue(LicenseVerifier.verifyLicense(generatedLicense, Files.readAllBytes(pubKeyPath)));
    }

    private static License generateSignedLicense(
            TestUtils.LicenseSpec spec, Path pubKeyPath, Path priKeyPath) throws Exception {
        LicenseSigner signer = new LicenseSigner(priKeyPath, pubKeyPath);
        License.Builder builder = License.builder()
                .uid(spec.uid)
                .feature(spec.feature)
                .type(spec.type)
                .subscriptionType(spec.subscriptionType)
                .issuedTo(spec.issuedTo)
                .issuer(spec.issuer)
                .maxNodes(spec.maxNodes);

        if (spec.expiryDate != null) {
            builder.expiryDate(DateUtils.endOfTheDay(spec.expiryDate));
        } else {
            builder.expiryDate(spec.expiryDateInMillis);
        }
        if (spec.issueDate != null) {
            builder.issueDate(DateUtils.beginningOfTheDay(spec.issueDate));
        } else {
            builder.issueDate(spec.issueDateInMillis);
        }
        builder.version(spec.version);
        return signer.sign(builder.build());
    }

}
```

```
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license.licensor;

import org.elasticsearch.common.Strings;
import org.elasticsearch.common.joda.DateMathParser;
import org.elasticsearch.common.joda.FormatDateTimeFormatter;
import org.elasticsearch.common.joda.Joda;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.DateUtils;
import org.elasticsearch.license.License;
import org.elasticsearch.test.ESTestCase;
import org.hamcrest.MatcherAssert;
import org.joda.time.format.DateTimeFormatter;

import java.io.IOException;
import java.nio.file.Path;
import java.util.UUID;

import static com.carrotsearch.randomizedtesting.RandomizedTest.randomBoolean;
import static com.carrotsearch.randomizedtesting.RandomizedTest.randomInt;
import static com.carrotsearch.randomizedtesting.RandomizedTest.randomIntBetween;
import static org.elasticsearch.common.xcontent.XContentFactory.jsonBuilder;
import static org.elasticsearch.test.ESTestCase.randomFrom;
import static org.hamcrest.core.IsEqual.equalTo;

public class TestUtils {

  public static final String PUBLIC_KEY_RESOURCE = "/public.key";
  public static final String PRIVATE_KEY_RESOURCE = "/private.key";

  private static final FormatDateTimeFormatter formatDateTimeFormatter =
      Joda.forPattern("yyyy-MM-dd");
  private static final DateMathParser dateMathParser =
      new DateMathParser(formatDateTimeFormatter);
  private static final DateTimeFormatter dateTimeFormatter = formatDateTimeFormatter.printer();

  public static String dumpLicense(License license) throws Exception {
    XContentBuilder builder = XContentFactory.contentBuilder(XContentType.JSON);
    builder.startObject();
    builder.startObject("license");
```

```
        license.toInnerXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.endObject();
        builder.endObject();
        return Strings.toString(builder);
    }

    public static String dateMathString(String time, final long now) {
        return dateTimeFormatter.print(dateMathParser.parse(time, () -> now));
    }

    public static long dateMath(String time, final long now) {
        return dateMathParser.parse(time, () -> now);
    }

    public static LicenseSpec generateRandomLicenseSpec(int version) {
        boolean datesInMillis = randomBoolean();
        long now = System.currentTimeMillis();
        String uid = UUID.randomUUID().toString();
        String issuer = "issuer__"  + randomInt();
        String issuedTo = "issuedTo__" + randomInt();
        String type = version < License.VERSION_NO_FEATURE_TYPE ?
                randomFrom("subscription", "internal", "development") :
                randomFrom("basic", "silver", "dev", "gold", "platinum");
        final String subscriptionType;
        final String feature;
        if (version < License.VERSION_NO_FEATURE_TYPE) {
            subscriptionType = randomFrom("gold", "silver", "platinum");
            feature = "feature__" + randomInt();
        } else {
            subscriptionType = null;
            feature = null;
        }
        int maxNodes = randomIntBetween(5, 100);
        if (datesInMillis) {
            long issueDateInMillis = dateMath("now", now);
            long expiryDateInMillis = dateMath("now+10d/d", now);
            return new LicenseSpec(
                    version,
                    uid,
                    feature,
                    issueDateInMillis,
                    expiryDateInMillis,
                    type,
                    subscriptionType,
                    issuedTo,
                    issuer,
                    maxNodes);
        } else {
```

```
        String issueDate = dateMathString("now", now);
        String expiryDate = dateMathString("now+10d/d", now);
        return new LicenseSpec(
                version,
                uid,
                feature,
                issueDate,
                expiryDate, type,
                subscriptionType,
                issuedTo,
                issuer,
                maxNodes);
    }
}

public static String generateLicenseSpecString(LicenseSpec licenseSpec) throws IOException {
    XContentBuilder licenses = jsonBuilder();
    licenses.startObject();
    licenses.startObject("license")
            .field("uid", licenseSpec.uid)
            .field("type", licenseSpec.type)
            .field("subscription_type", licenseSpec.subscriptionType)
            .field("issued_to", licenseSpec.issuedTo)
            .field("issuer", licenseSpec.issuer)
            .field("feature", licenseSpec.feature)
            .field("max_nodes", licenseSpec.maxNodes);

    if (licenseSpec.issueDate != null) {
        licenses.field("issue_date", licenseSpec.issueDate);
    } else {
        licenses.field("issue_date_in_millis", licenseSpec.issueDateInMillis);
    }
    if (licenseSpec.expiryDate != null) {
        licenses.field("expiry_date", licenseSpec.expiryDate);
    } else {
        licenses.field("expiry_date_in_millis", licenseSpec.expiryDateInMillis);
    }
    licenses.field("version", licenseSpec.version);
    licenses.endObject();
    licenses.endObject();
    return Strings.toString(licenses);
}

public static void assertLicenseSpec(LicenseSpec spec, License license) {
    MatcherAssert.assertThat(license.uid(), equalTo(spec.uid));
    MatcherAssert.assertThat(license.issuedTo(), equalTo(spec.issuedTo));
    MatcherAssert.assertThat(license.issuer(), equalTo(spec.issuer));
    MatcherAssert.assertThat(license.type(), equalTo(spec.type));
```

```
      MatcherAssert.assertThat(license.maxNodes(), equalTo(spec.maxNodes));
      if (spec.issueDate != null) {
        MatcherAssert.assertThat(
            license.issueDate(),
            equalTo(DateUtils.beginningOfTheDay(spec.issueDate)));
      } else {
        MatcherAssert.assertThat(license.issueDate(), equalTo(spec.issueDateInMillis));
      }
      if (spec.expiryDate != null) {
        MatcherAssert.assertThat(
            license.expiryDate(),
            equalTo(DateUtils.endOfTheDay(spec.expiryDate)));
      } else {
        MatcherAssert.assertThat(license.expiryDate(), equalTo(spec.expiryDateInMillis));
      }
    }

    public static License generateSignedLicense(
        TimeValue expiryDuration, Path pubKeyPath, Path priKeyPath) throws Exception {
      long issue = System.currentTimeMillis();
      int version = ESTestCase.randomIntBetween(License.VERSION_START, License.VERSION_CURRENT);
      String type = version < License.VERSION_NO_FEATURE_TYPE ?
          randomFrom("subscription", "internal", "development") :
          randomFrom("trial", "basic", "silver", "dev", "gold", "platinum");
      final License.Builder builder = License.builder()
          .uid(UUID.randomUUID().toString())
          .expiryDate(issue + expiryDuration.getMillis())
          .issueDate(issue)
          .version(version)
          .type(type)
          .issuedTo("customer")
          .issuer("elasticsearch")
          .maxNodes(5);
      if (version == License.VERSION_START) {
        builder.subscriptionType(randomFrom("dev", "gold", "platinum", "silver"));
        builder.feature(ESTestCase.randomAlphaOfLength(10));
      }
      LicenseSigner signer = new LicenseSigner(priKeyPath, pubKeyPath);
      return signer.sign(builder.build());
    }

    public static class LicenseSpec {
      public final int version;
      public final String feature;
      public final String issueDate;
      public final long issueDateInMillis;
      public final String expiryDate;
      public final long expiryDateInMillis;
```

```
public final String uid;
public final String type;
public final String subscriptionType;
public final String issuedTo;
public final String issuer;
public final int maxNodes;

public LicenseSpec(
        int version,
        String uid,
        String feature,
        long issueDateInMillis,
        long expiryDateInMillis,
        String type,
        String subscriptionType,
        String issuedTo,
        String issuer,
        int maxNodes) {
    this.version = version;
    this.feature = feature;
    this.issueDateInMillis = issueDateInMillis;
    this.issueDate = null;
    this.expiryDateInMillis = expiryDateInMillis;
    this.expiryDate = null;
    this.uid = uid;
    this.type = type;
    this.subscriptionType = subscriptionType;
    this.issuedTo = issuedTo;
    this.issuer = issuer;
    this.maxNodes = maxNodes;
}

public LicenseSpec(
        int version,
        String uid,
        String feature,
        String issueDate,
        String expiryDate,
        String type,
        String subscriptionType,
        String issuedTo,
        String issuer,
        int maxNodes) {
    this.version = version;
    this.feature = feature;
    this.issueDate = issueDate;
    this.issueDateInMillis = -1;
    this.expiryDate = expiryDate;
```

```
            this.expiryDateInMillis = -1;
            this.uid = uid;
            this.type = type;
            this.subscriptionType = subscriptionType;
            this.issuedTo = issuedTo;
            this.issuer = issuer;
            this.maxNodes = maxNodes;
        }
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license.licensor.tools;

import org.elasticsearch.cli.Command;
import org.elasticsearch.cli.CommandTestCase;
import org.elasticsearch.cli.ExitCodes;
import org.elasticsearch.cli.UserException;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.license.License;
import org.elasticsearch.license.licensor.TestUtils;
import org.junit.Before;

import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;

public class LicenseVerificationToolTests extends CommandTestCase {
    protected Path pubKeyPath = null;
    protected Path priKeyPath = null;

    @Before
    public void setup() throws Exception {
        logger.error("project.basedir [{}]", System.getProperty("project.basedir"));
        pubKeyPath = getDataPath(TestUtils.PUBLIC_KEY_RESOURCE);
        priKeyPath = getDataPath(TestUtils.PRIVATE_KEY_RESOURCE);
    }

    @Override
    protected Command newCommand() {
        return new LicenseVerificationTool();
    }

    public void testMissingKeyPath() throws Exception {
```

```
    Path pub = createTempDir().resolve("pub");
    UserException e = expectThrows(
            UserException.class,
            () -> execute("--publicKeyPath", pub.toString()));
    assertTrue(e.getMessage(), e.getMessage().contains("pub does not exist"));
    assertEquals(ExitCodes.USAGE, e.exitCode);
}

public void testMissingLicenseSpec() throws Exception {
    UserException e = expectThrows(UserException.class, () -> {
        execute("--publicKeyPath", pubKeyPath.toString());
    });
    assertTrue(
            e.getMessage(),
            e.getMessage().contains("Must specify either --license or --licenseFile"));
    assertEquals(ExitCodes.USAGE, e.exitCode);
}

public void testBrokenLicense() throws Exception {
    final TimeValue oneHour = TimeValue.timeValueHours(1);
    License signedLicense = TestUtils.generateSignedLicense(oneHour, pubKeyPath, priKeyPath);
    License tamperedLicense = License.builder()
        .fromLicenseSpec(signedLicense, signedLicense.signature())
        .expiryDate(signedLicense.expiryDate() + randomIntBetween(1, 1000)).build();
    UserException e = expectThrows(
            UserException.class,
            () -> execute(
                    "--publicKeyPath",
                    pubKeyPath.toString(),
                    "--license",
                    TestUtils.dumpLicense(tamperedLicense)));
    assertEquals("Invalid License!", e.getMessage());
    assertEquals(ExitCodes.DATA_ERROR, e.exitCode);
}

public void testLicenseSpecString() throws Exception {
    final TimeValue oneHour = TimeValue.timeValueHours(1);
    License signedLicense = TestUtils.generateSignedLicense(oneHour, pubKeyPath, priKeyPath);
    String output = execute(
            "--publicKeyPath",
            pubKeyPath.toString(),
            "--license",
            TestUtils.dumpLicense(signedLicense));
    assertFalse(output, output.isEmpty());
}

public void testLicenseSpecFile() throws Exception {
    final TimeValue oneHour = TimeValue.timeValueHours(1);
    License signedLicense = TestUtils.generateSignedLicense(oneHour, pubKeyPath, priKeyPath);
    Path licenseSpecFile = createTempFile();
    Files.write(
        licenseSpecFile,
        TestUtils.dumpLicense(signedLicense).getBytes(StandardCharsets.UTF_8));
    String output = execute(
        "--publicKeyPath",
        pubKeyPath.toString(),
        "--licenseFile",
        licenseSpecFile.toString());
    assertFalse(output, output.isEmpty());
  }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license.licensor.tools;

import org.elasticsearch.cli.Command;
import org.elasticsearch.cli.CommandTestCase;
import org.elasticsearch.cli.ExitCodes;
import org.elasticsearch.cli.UserException;
import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.License;
import org.elasticsearch.license.licensor.TestUtils;
import org.junit.Before;

import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;

public class LicenseGenerationToolTests extends CommandTestCase {

  protected Path pubKeyPath = null;
  protected Path priKeyPath = null;

  @Before
  public void setup() throws Exception {
    pubKeyPath = getDataPath(TestUtils.PUBLIC_KEY_RESOURCE);
    priKeyPath = getDataPath(TestUtils.PRIVATE_KEY_RESOURCE);
  }

  @Override
  protected Command newCommand() {
    return new LicenseGeneratorTool();
}

public void testMissingKeyPaths() throws Exception {
    Path pub = createTempDir().resolve("pub");
    Path pri = createTempDir().resolve("pri");
    UserException e = expectThrows(
            UserException.class,
            () -> execute(
                    "--publicKeyPath",
                    pub.toString(),
                    "--privateKeyPath",
                    pri.toString()));
    assertTrue(e.getMessage(), e.getMessage().contains("pri does not exist"));
    assertEquals(ExitCodes.USAGE, e.exitCode);

    Files.createFile(pri);
    e = expectThrows(
            UserException.class,
            () -> execute(
                    "--publicKeyPath",
                    pub.toString(),
                    "--privateKeyPath",
                    pri.toString()));
    assertTrue(e.getMessage(), e.getMessage().contains("pub does not exist"));
    assertEquals(ExitCodes.USAGE, e.exitCode);
}

public void testMissingLicenseSpec() throws Exception {
    UserException e = expectThrows(
            UserException.class,
            () -> execute(
                    "--publicKeyPath",
                    pubKeyPath.toString(),
                    "--privateKeyPath",
                    priKeyPath.toString()));
    assertTrue(
            e.getMessage(),
            e.getMessage().contains("Must specify either --license or --licenseFile"));
    assertEquals(ExitCodes.USAGE, e.exitCode);
}

public void testLicenseSpecString() throws Exception {
    TestUtils.LicenseSpec inputLicenseSpec =
            TestUtils.generateRandomLicenseSpec(License.VERSION_CURRENT);
    String licenseSpecString = TestUtils.generateLicenseSpecString(inputLicenseSpec);
    String output = execute(
            "--publicKeyPath",
                pubKeyPath.toString(),
                "--privateKeyPath",
                priKeyPath.toString(),
                "--license",
                licenseSpecString);
        final BytesArray bytes = new BytesArray(output.getBytes(StandardCharsets.UTF_8));
        License outputLicense = License.fromSource(bytes, XContentType.JSON);
        TestUtils.assertLicenseSpec(inputLicenseSpec, outputLicense);
    }

    public void testLicenseSpecFile() throws Exception {
        TestUtils.LicenseSpec inputLicenseSpec =
                TestUtils.generateRandomLicenseSpec(License.VERSION_CURRENT);
        String licenseSpecString = TestUtils.generateLicenseSpecString(inputLicenseSpec);
        Path licenseSpecFile = createTempFile();
        Files.write(licenseSpecFile, licenseSpecString.getBytes(StandardCharsets.UTF_8));
        String output = execute(
                "--publicKeyPath",
                pubKeyPath.toString(),
                "--privateKeyPath",
                priKeyPath.toString(),
                "--licenseFile",
                licenseSpecFile.toString());
        final BytesArray bytes = new BytesArray(output.getBytes(StandardCharsets.UTF_8));
        License outputLicense = License.fromSource(bytes, XContentType.JSON);
        TestUtils.assertLicenseSpec(inputLicenseSpec, outputLicense);
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license.licensor.tools;

import org.elasticsearch.cli.Command;
import org.elasticsearch.cli.CommandTestCase;
import org.elasticsearch.cli.ExitCodes;
import org.elasticsearch.cli.UserException;

import java.nio.file.Files;
import java.nio.file.Path;

import static org.hamcrest.CoreMatchers.containsString;

public class KeyPairGenerationToolTests extends CommandTestCase {
    @Override
    protected Command newCommand() {
        return new KeyPairGeneratorTool();
    }

    public void testMissingKeyPaths() throws Exception {
        Path exists = createTempFile("", "existing");
        Path dne = createTempDir().resolve("dne");
        UserException e = expectThrows(
                UserException.class,
                () -> execute(
                        "--publicKeyPath",
                        exists.toString(),
                        "--privateKeyPath",
                        dne.toString()));
        assertThat(e.getMessage(), containsString("existing"));
        assertEquals(ExitCodes.USAGE, e.exitCode);
        e = expectThrows(
                UserException.class,
                () -> execute(
                        "--publicKeyPath",
                        dne.toString(),
                        "--privateKeyPath",
                        exists.toString()));
        assertThat(e.getMessage(), containsString("existing"));
        assertEquals(ExitCodes.USAGE, e.exitCode);
    }

    public void testTool() throws Exception {
        Path keysDir = createTempDir();
        Path publicKeyFilePath = keysDir.resolve("public");
        Path privateKeyFilePath = keysDir.resolve("private");

        execute(
                "--publicKeyPath",
                publicKeyFilePath.toString(),
                "--privateKeyPath",
                privateKeyFilePath.toString());
        assertTrue(publicKeyFilePath.toString(), Files.exists(publicKeyFilePath));
        assertTrue(privateKeyFilePath.toString(), Files.exists(privateKeyFilePath));
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
```

```
package org.elasticsearch.license.licensor;

import org.apache.lucene.util.BytesRef;
import org.apache.lucene.util.BytesRefIterator;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.CryptUtils;
import org.elasticsearch.license.License;

import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.file.Files;
import java.nio.file.Path;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.Signature;
import java.security.SignatureException;
import java.util.Base64;
import java.util.Collections;
import java.util.Map;

/**
* Responsible for generating a license signature according to the signature spec and sign it with
* the provided encrypted private key
*/
public class LicenseSigner {

   private static final int MAGIC_LENGTH = 13;

   private final Path publicKeyPath;

   private final Path privateKeyPath;

   public LicenseSigner(final Path privateKeyPath, final Path publicKeyPath) {
       this.publicKeyPath = publicKeyPath;
       this.privateKeyPath = privateKeyPath;
   }

   /**
    * Generates a signature for the {@code licenseSpec}. Signature structure:
    * <code>
    * | VERSION | MAGIC | PUB_KEY_DIGEST | SIGNED_LICENSE_CONTENT |
    * </code>
    *
```

```
     * @return a signed License
     */
    public License sign(License licenseSpec) throws IOException {
        XContentBuilder contentBuilder = XContentFactory.contentBuilder(XContentType.JSON);
        final Map<String, String> licenseSpecViewMode =
                Collections.singletonMap(License.LICENSE_SPEC_VIEW_MODE, "true");
        licenseSpec.toXContent(contentBuilder, new ToXContent.MapParams(licenseSpecViewMode));
        final byte[] signedContent;
        try {
            final Signature rsa = Signature.getInstance("SHA512withRSA");
            rsa.initSign(CryptUtils.readEncryptedPrivateKey(Files.readAllBytes(privateKeyPath)));
            final BytesRefIterator iterator = BytesReference.bytes(contentBuilder).iterator();
            BytesRef ref;
            while((ref = iterator.next()) != null) {
                rsa.update(ref.bytes, ref.offset, ref.length);
            }
            signedContent = rsa.sign();
        } catch (InvalidKeyException
                | IOException
                | NoSuchAlgorithmException
                | SignatureException e) {
            throw new IllegalStateException(e);
        }
        final byte[] magic = new byte[MAGIC_LENGTH];
        SecureRandom random = new SecureRandom();
        random.nextBytes(magic);
        final byte[] hash = Base64.getEncoder().encode(Files.readAllBytes(publicKeyPath));
        assert hash != null;
        byte[] bytes = new byte[4 + 4 + MAGIC_LENGTH + 4 + hash.length + 4 + signedContent.length];
        ByteBuffer byteBuffer = ByteBuffer.wrap(bytes);
        byteBuffer.putInt(licenseSpec.version())
                .putInt(magic.length)
                .put(magic)
                .putInt(hash.length)
                .put(hash)
                .putInt(signedContent.length)
                .put(signedContent);

        return License.builder()
                .fromLicenseSpec(licenseSpec, Base64.getEncoder().encodeToString(bytes))
                .build();
    }
}
```

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

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

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

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

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

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

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

1.7. "License" means this document.

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

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

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

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

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

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

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

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

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

2. License Grants.

2.1. The Initial Developer Grant.

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

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

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

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

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

2.2. Contributor Grant.

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

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

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

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

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

3. Distribution Obligations.

3.1. Availability of Source Code.

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

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You

believe Your Modifications are Your original creation(s) and/or You
have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

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

3.4. Application of Additional Terms.

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

3.5. Distribution of Executable Versions.

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

3.6. Larger Works.

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

4. Versions of the License.

4.1. New Versions.

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

4.2. Effect of New Versions.

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

4.3. Modified Versions.

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

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN

ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

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

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

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

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

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE

TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

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

9. MISCELLANEOUS.

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

10. RESPONSIBILITY FOR CLAIMS.

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

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

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

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

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

Preamble

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

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous

contributions to the wide range of software distributed through that
system in reliance on consistent application of that system; it is up to
the author/donor to decide if he or she is willing to distribute
software through any other system and a licensee cannot impose that choice.

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

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

```
One line to give the program's name and a brief idea of what it does.
Copyright (C) <year> <name of author>

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

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

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA
```

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

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
```

`show w'. This is free software, and you are welcome to redistribute
it under certain conditions; type `show c' for details.

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

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

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

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

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

#

Certain source files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

You should also note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with

Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result. Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Proceed with caution and we recommend that you obtain the advice of a lawyer skilled in open source matters before removing the Classpath Exception or making modifications to this package which may subsequently be redistributed and/or involve the use of third party software.

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

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.
You may use under either the Apache License Version 2.0 or the BSD 3-Clause License.

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Copyright (c) 2011, Mike Samuel
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.

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.
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

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

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

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

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

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

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

1.7. License means this document.

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

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

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

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

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

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

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

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

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

2. License Grants.

2.1. The Initial Developer Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section2.1(b) above, no patent license is granted: (1)for code that You delete from the Original Software, or (2)for infringements caused by: (i)the modification of the Original Software, or (ii)the combination of the Original Software with other software or devices.

2.2. Contributor Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce,

modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1)Modifications made by that Contributor (or portions thereof); and (2)the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) The licenses granted in Sections2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.
(d) Notwithstanding Section2.2(b) above, no patent license is granted: (1)for any code that Contributor has deleted from the Contributor Version; (2)for infringements caused by: (i)third party modifications of Contributor Version, or (ii)the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3)under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

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

3.2. Modifications.

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

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

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

3.5. Distribution of Executable Versions.

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

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

4. Versions of the License.

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

4.2. Effect of New Versions.

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

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

5. DISCLAIMER OF WARRANTY.

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

6. TERMINATION.

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

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

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

8. U.S. GOVERNMENT END USERS.

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

9. MISCELLANEOUS.

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

10. RESPONSIBILITY FOR CLAIMS.

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

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

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

===

This project includes Public Suffix List copied from
<https://publicsuffix.org/list/effective_tld_names.dat>
licensed under the terms of the Mozilla Public License, v. 2.0

Full license text: <http://mozilla.org/MPL/2.0/>

Mozilla Public License Version 2.0
==================================

1. Definitions

1.1. "Contributor"
means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

1.2. "Contributor Version"
means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
means Covered Software of a particular Contributor.

1.4. "Covered Software"
means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

1.5. "Incompatible With Secondary Licenses"
means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of
version 1.1 or earlier of the License, but not also under the
terms of a Secondary License.

1.6. "Executable Form"
means any form of the work other than Source Code Form.

1.7. "Larger Work"
means a work that combines Covered Software with other material, in
a separate file or files, that is not Covered Software.

1.8. "License"
means this document.

1.9. "Licensable"
means having the right to grant, to the maximum extent possible,
whether at the time of the initial grant or subsequently, any and
all of the rights conveyed by this License.

1.10. "Modifications"
means any of the following:

(a) any file in Source Code Form that results from an addition to,
deletion from, or modification of the contents of Covered
Software; or

(b) any new file in Source Code Form that contains any Covered
Software.

1.11. "Patent Claims" of a Contributor
means any patent claim(s), including without limitation, method,
process, and apparatus claims, in any patent Licensable by such
Contributor that would be infringed, but for the grant of the
License, by the making, using, selling, offering for sale, having
made, import, or transfer of either its Contributions or its
Contributor Version.

1.12. "Secondary License"
means either the GNU General Public License, Version 2.0, the GNU
Lesser General Public License, Version 2.1, the GNU Affero General
Public License, Version 3.0, or any later versions of those
licenses.

1.13. "Source Code Form"
means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

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

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of
its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code
Form, as described in Section 3.1, and You must inform recipients of
the Executable Form how they can obtain a copy of such Source Code
Form by reasonable means in a timely manner, at a charge no more
than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this
License, or sublicense it under different terms, provided that the
license for the Executable Form does not attempt to limit or alter
the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the      *
*  Covered Software is free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*                                                                      *
*  Under no circumstances and under no legal theory, whether tort      *
*  (including negligence), contract, or otherwise, shall any           *
*  Contributor, or anyone who distributes Covered Software as          *
*  permitted above, be liable to You for any direct, indirect,         *
*  special, incidental, or consequential damages of any character      *
*  including, without limitation, damages for lost profits, loss of    *
*  goodwill, work stoppage, computer failure or malfunction, or any    *
*  and all other commercial damages or losses, even if such party      *
*  shall have been informed of the possibility of such damages. This   *
*  limitation of liability shall not apply to liability for death or   *
*  personal injury resulting from such party's negligence to the       *
*  extent applicable law prohibits such limitation. Some               *
*  jurisdictions do not allow the exclusion or limitation of           *
*  incidental or consequential damages, so this exclusion and          *
*  limitation may not apply to You.                                    *
*                                                                      *
************************************************************************
```

8. Litigation

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

This Source Code Form is "Incompatible With Secondary Licenses", as
defined by the Mozilla Public License, v. 2.0.

UnboundID LDAP SDK Free Use License

THIS IS AN AGREEMENT BETWEEN YOU ("YOU") AND UNBOUNDID CORP. ("UNBOUNDID") REGARDING YOUR USE OF UNBOUNDID LDAP SDK FOR JAVA AND ANY ASSOCIATED DOCUMENTATION, OBJECT CODE, COMPILED LIBRARIES, SOURCE CODE AND SOURCE FILES OR OTHER MATERIALS MADE AVAILABLE BY UNBOUNDID (COLLECTIVELY REFERRED TO IN THIS AGREEMENT AS THE ("SDK").

BY INSTALLING, ACCESSING OR OTHERWISE USING THE SDK, YOU ACCEPT THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT INSTALL, ACCESS OR USE THE SDK.

USE OF THE SDK. Subject to your compliance with this Agreement, UnboundID grants to You a non-exclusive, royalty-free license, under UnboundID's intellectual property rights in the SDK, to use, reproduce, modify and distribute this release of the SDK; provided that no license is granted herein under any patents that may be infringed by your modifications, derivative works or by other works in which the SDK may be incorporated (collectively, your "Applications"). You may reproduce and redistribute the SDK with your Applications provided that you (i) include this license file and an unmodified copy of the unboundid-ldapsdk-se.jar file; and (ii) such redistribution is subject to a license whose terms do not conflict with or contradict the terms of this Agreement. You may also reproduce and redistribute the SDK without your Applications provided that you redistribute the SDK complete and unmodified (i.e., with all "read me" files, copyright notices, and other legal notices and terms that UnboundID has included in the SDK).

SCOPE OF LICENSES. This Agreement does not grant You the right to use any UnboundID intellectual property which is not included as part of the SDK. The SDK is licensed, not sold. This Agreement only gives You some rights to use the SDK. UnboundID reserves all other rights. Unless applicable law gives You more rights despite this limitation, You may use the SDK only as expressly permitted in this Agreement.

SUPPORT. UnboundID is not obligated to provide any technical or other support ("Support Services") for the SDK to You under this Agreement. However, if UnboundID chooses to provide any Support Services to You, Your use of such Support Services will be governed by then-current UnboundID support policies.

TERMINATION. UnboundID reserves the right to discontinue offering the SDK and
to modify the SDK at any time in its sole discretion. Notwithstanding anything
contained in this Agreement to the contrary, UnboundID may also, in its sole
discretion, terminate or suspend access to the SDK to You or any end user at
any time. In addition, if you fail to comply with the terms of this Agreement,
then any rights granted herein will be automatically terminated if such failure
is not corrected within 30 days of the initial notification of such failure.
You acknowledge that termination and/or monetary damages may not be a
sufficient remedy if You breach this Agreement and that UnboundID will be
entitled, without waiving any other rights or remedies, to injunctive or
equitable relief as may be deemed proper by a court of competent jurisdiction
in the event of a breach. UnboundID may also terminate this Agreement if the
SDK becomes, or in UnboundID?s reasonable opinion is likely to become, the
subject of a claim of intellectual property infringement or trade secret
misappropriation. All rights and licenses granted herein will simultaneously
and automatically terminate upon termination of this Agreement for any reason.

DISCLAIMER OF WARRANTY. THE SDK IS PROVIDED "AS IS" AND UNBOUNDID DOES NOT
WARRANT THAT THE SDK WILL BE ERROR-FREE, VIRUS-FREE, WILL PERFORM IN AN
UNINTERRUPTED, SECURE OR TIMELY MANNER, OR WILL INTEROPERATE WITH OTHER
HARDWARE, SOFTWARE, SYSTEMS OR DATA. TO THE MAXIMUM EXTENT ALLOWED BY LAW,
ALL
CONDITIONS, REPRESENTATIONS AND WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY
OR OTHERWISE INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (EVEN IF UNBOUNDID HAD BEEN
INFORMED OF SUCH PURPOSE), OR NON-INFRINGEMENT OF THIRD PARTY RIGHTS ARE HEREBY
DISCLAIMED.

LIMITATION OF LIABILITY. IN NO EVENT WILL UNBOUNDID OR ITS SUPPLIERS BE LIABLE
FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS,
REVENUE, DATA OR DATA USE, BUSINESS INTERRUPTION, COST OF COVER, DIRECT,
INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND)
ARISING OUT OF THE USE OF OR INABILITY TO USE THE SDK OR IN ANY WAY RELATED TO
THIS AGREEMENT, EVEN IF UNBOUNDID HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

ADDITIONAL RIGHTS. Certain states do not allow the exclusion of implied
warranties or limitation of liability for certain kinds of damages, so the
exclusion of limited warranties and limitation of liability set forth above may
not apply to You.

EXPORT RESTRICTIONS. The SDK is subject to United States export control laws.
You acknowledge and agree that You are responsible for compliance with all
domestic and international export laws and regulations that apply to the SDK.

MISCELLANEOUS. This Agreement constitutes the entire agreement with respect to
the SDK. If any provision of this Agreement shall be held to be invalid,
illegal or unenforceable, the validity, legality and enforceability of the

remaining provisions shall in no way be affected or impaired thereby. This Agreement and performance hereunder shall be governed by and construed in accordance with the laws of the State of Texas without regard to its conflict of laws rules. Any disputes related to this Agreement shall be exclusively litigated in the state or federal courts located in Travis County, Texas.
Apache Commons Codec
Copyright 2002-2014 The Apache Software Foundation

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

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.
Copyright (c) 2000-2015 The Legion of the Bouncy Castle Inc. (http://www.bouncycastle.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.
Apache HttpComponents Client
Copyright 1999-2016 The Apache Software Foundation

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

```
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.test.ESTestCase;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;

public class FieldPermissionsCacheTests extends ESTestCase {

  public void testFieldPermissionsCaching() {
    FieldPermissionsCache fieldPermissionsCache = new FieldPermissionsCache(Settings.EMPTY);
    String[] allowed = new String[]{randomAlphaOfLength(5) + "*", randomAlphaOfLength(5) + "*",
randomAlphaOfLength(5) + "*"};
    String[] denied = new String[]{allowed[0] + randomAlphaOfLength(5), allowed[1] +
randomAlphaOfLength(5),
        allowed[2] + randomAlphaOfLength(5)};
    FieldPermissions fieldPermissions = fieldPermissionsCache.getFieldPermissions(allowed, denied);
    assertNotNull(fieldPermissions);
    final String[] allowed2 = randomBoolean() ? allowed : Arrays.copyOf(allowed, allowed.length);
    final String[] denied2 = randomBoolean() ? denied : Arrays.copyOf(denied, denied.length);
    assertSame(fieldPermissions, fieldPermissionsCache.getFieldPermissions(allowed2, denied2));
  }

  public void testMergeFieldPermissions() {
    FieldPermissionsCache fieldPermissionsCache = new FieldPermissionsCache(Settings.EMPTY);
    String allowedPrefix1 = randomAlphaOfLength(5);
    String allowedPrefix2 = randomAlphaOfLength(5);
    String[] allowed1 = new String[]{allowedPrefix1 + "*"};
    String[] allowed2 = new String[]{allowedPrefix2 + "*"};
    String[] denied1 = new String[]{allowedPrefix1 + "a"};
    String[] denied2 = new String[]{allowedPrefix2 + "a"};
    FieldPermissions fieldPermissions1 = randomBoolean() ?
fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
        new FieldPermissions(fieldPermissionDef(allowed1, denied1));
    FieldPermissions fieldPermissions2 = randomBoolean() ?
fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
        new FieldPermissions(fieldPermissionDef(allowed2, denied2));
    FieldPermissions mergedFieldPermissions =
        fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
    assertTrue(mergedFieldPermissions.grantsAccessTo(allowedPrefix1 + "b"));
    assertTrue(mergedFieldPermissions.grantsAccessTo(allowedPrefix2 + "b"));
    assertFalse(mergedFieldPermissions.grantsAccessTo(denied1[0]));
    assertFalse(mergedFieldPermissions.grantsAccessTo(denied2[0]));

    allowed1 = new String[]{randomAlphaOfLength(5) + "*", randomAlphaOfLength(5) + "*"};
    allowed2 = null;
    denied1 = new String[]{allowed1[0] + "a", allowed1[1] + "a"};
    denied2 = null;
    fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
        new FieldPermissions(fieldPermissionDef(allowed1, denied1));
    fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
        new FieldPermissions(fieldPermissionDef(allowed2, denied2));
    mergedFieldPermissions =
        fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
    assertFalse(mergedFieldPermissions.hasFieldLevelSecurity());

    allowed1 = new String[]{};
    allowed2 = new String[]{randomAlphaOfLength(5) + "*", randomAlphaOfLength(5) + "*"};
    denied1 = new String[]{};
    denied2 = new String[]{allowed2[0] + "a", allowed2[1] + "a"};
    fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
        new FieldPermissions(fieldPermissionDef(allowed1, denied1));
    fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
        new FieldPermissions(fieldPermissionDef(allowed2, denied2));
    mergedFieldPermissions =
        fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
    for (String field : allowed2) {
        assertTrue(mergedFieldPermissions.grantsAccessTo(field));
    }
    for (String field : denied2) {
        assertFalse(mergedFieldPermissions.grantsAccessTo(field));
    }

    allowed1 = randomBoolean() ? null : new String[]{"*"};
    allowed2 = randomBoolean() ? null : new String[]{"*"};
    denied1 = new String[]{"a"};
    denied2 = new String[]{"b"};
    fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
        new FieldPermissions(fieldPermissionDef(allowed1, denied1));
    fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
        new FieldPermissions(fieldPermissionDef(allowed2, denied2));
    mergedFieldPermissions =
        fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
    assertTrue(mergedFieldPermissions.grantsAccessTo("a"));
    assertTrue(mergedFieldPermissions.grantsAccessTo("b"));

    // test merge does not remove _all
    allowed1 = new String[]{"_all"};
```

```
        allowed2 = new String[]{};
        denied1 = null;
        denied2 = null;
        fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
                new FieldPermissions(fieldPermissionDef(allowed1, denied1));
        fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
                new FieldPermissions(fieldPermissionDef(allowed2, denied2));
        mergedFieldPermissions =
                fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
        assertTrue(fieldPermissions1.grantsAccessTo("_all"));
        assertFalse(fieldPermissions2.grantsAccessTo("_all"));
        assertTrue(mergedFieldPermissions.grantsAccessTo("_all"));

        allowed1 = new String[] { "a*" };
        allowed2 = new String[] { "b*" };
        denied1 = new String[] { "aa*" };
        denied2 = null;
        fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
                new FieldPermissions(fieldPermissionDef(allowed1, denied1));
        fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
                new FieldPermissions(fieldPermissionDef(allowed2, denied2));
        mergedFieldPermissions =
                fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
        assertTrue(mergedFieldPermissions.grantsAccessTo("a"));
        assertTrue(mergedFieldPermissions.grantsAccessTo("b"));
        assertFalse(mergedFieldPermissions.grantsAccessTo("aa"));
        assertFalse(mergedFieldPermissions.grantsAccessTo("aa1"));
        assertTrue(mergedFieldPermissions.grantsAccessTo("a1"));
    }

    public void testNonFlsAndFlsMerging() {
        List<FieldPermissions> permissionsList = new ArrayList<>();
        permissionsList.add(new FieldPermissions(fieldPermissionDef(new String[] {"field1"}, null)));
        permissionsList.add(new FieldPermissions(fieldPermissionDef(new String[] {"field2", "query*"}, null)));
        permissionsList.add(new FieldPermissions(fieldPermissionDef(new String[] {"field1", "field2"}, null)));
        permissionsList.add(new FieldPermissions(fieldPermissionDef(new String[] {}, null)));
        permissionsList.add(new FieldPermissions(fieldPermissionDef(null, null)));

        FieldPermissionsCache cache = new FieldPermissionsCache(Settings.EMPTY);
        for (int i = 0; i < scaledRandomIntBetween(1, 12); i++) {
            Collections.shuffle(permissionsList, random());
            FieldPermissions result = cache.getFieldPermissions(permissionsList);
            assertFalse(result.hasFieldLevelSecurity());
        }
    }

    private static FieldPermissionsDefinition fieldPermissionDef(String[] granted, String[] denied) {
        return new FieldPermissionsDefinition(granted, denied);
```

```
   }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.analysis.common.CommonAnalysisPlugin;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackClientPlugin;
import org.elasticsearch.xpack.core.XPackSettings;

import java.util.Arrays;
import java.util.Collection;
import java.util.concurrent.CountDownLatch;

public abstract class AbstractLicensesIntegrationTestCase extends ESIntegTestCase {

  @Override
  protected Settings nodeSettings(int nodeOrdinal) {
     return Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), false).build();
  }

  @Override
  protected Collection<Class<? extends Plugin>> nodePlugins() {
     return Arrays.asList(LocalStateCompositeXPackPlugin.class, CommonAnalysisPlugin.class);
  }

  @Override
  protected Collection<Class<? extends Plugin>> transportClientPlugins() {
     return Arrays.asList(XPackClientPlugin.class, CommonAnalysisPlugin.class);
  }

  @Override
  protected Settings transportClientSettings() {
     // Plugin should be loaded on the transport client as well
     return nodeSettings(0);
  }
  protected void putLicense(final License license) throws InterruptedException {
    final CountDownLatch latch = new CountDownLatch(1);
    ClusterService clusterService = internalCluster().getInstance(ClusterService.class,
internalCluster().getMasterName());
    clusterService.submitStateUpdateTask("putting license", new ClusterStateUpdateTask() {
      @Override
      public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
        latch.countDown();
      }

      @Override
      public ClusterState execute(ClusterState currentState) throws Exception {
        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
        mdBuilder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(license, null));
        return ClusterState.builder(currentState).metaData(mdBuilder).build();
      }

      @Override
      public void onFailure(String source, @Nullable Exception e) {
        logger.error("error on metaData cleanup after test", e);
      }
    });
    latch.await();
  }

  protected void putLicenseTombstone() throws InterruptedException {
    putLicense(LicensesMetaData.LICENSE_TOMBSTONE);
  }

  protected void wipeAllLicenses() throws InterruptedException {
    final CountDownLatch latch = new CountDownLatch(1);
    ClusterService clusterService = internalCluster().getInstance(ClusterService.class,
internalCluster().getMasterName());
    clusterService.submitStateUpdateTask("delete licensing metadata", new ClusterStateUpdateTask() {
      @Override
      public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
        latch.countDown();
      }

      @Override
      public ClusterState execute(ClusterState currentState) throws Exception {
        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
        mdBuilder.removeCustom(LicensesMetaData.TYPE);
        return ClusterState.builder(currentState).metaData(mdBuilder).build();
      }

      @Override
```

```
            public void onFailure(String source, @Nullable Exception e) {
                logger.error("error on metaData cleanup after test", e);
            }
        });
        latch.await();
    }

    protected void assertLicenseActive(boolean active) throws InterruptedException {
        boolean success = awaitBusy(() -> {
            for (XPackLicenseState licenseState : internalCluster().getDataNodeInstances(XPackLicenseState.class)) {
                if (licenseState.isActive() == active) {
                    return true;
                }
            }
            return false;
        });
        assertTrue(success);
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.cluster.ClusterChangedEvent;
import org.elasticsearch.cluster.ClusterName;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.junit.After;
import org.junit.Before;
import org.mockito.ArgumentCaptor;

import static java.util.Collections.emptyMap;
import static java.util.Collections.emptySet;
import static org.hamcrest.Matchers.equalTo;
import static org.mockito.Matchers.any;
import static org.mockito.Mockito.times;
import static org.mockito.Mockito.verify;
import static org.mockito.Mockito.when;
```

```
public class LicenseClusterChangeTests extends AbstractLicenseServiceTestCase {

  private TestUtils.AssertingLicenseState licenseState;

  @Before
  public void setup() {
     licenseState = new TestUtils.AssertingLicenseState();
     setInitialState(null, licenseState, Settings.EMPTY);
     licenseService.start();
  }

  @After
  public void teardown() {
     licenseService.stop();
  }


  public void testNotificationOnNewLicense() throws Exception {
     ClusterState oldState = ClusterState.builder(new ClusterName("a")).build();
     final License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(24));
     MetaData metaData = MetaData.builder().putCustom(LicensesMetaData.TYPE, new
LicensesMetaData(license, null)).build();
     ClusterState newState = ClusterState.builder(new ClusterName("a")).metaData(metaData).build();
     licenseService.clusterChanged(new ClusterChangedEvent("simulated", newState, oldState));
     assertThat(licenseState.activeUpdates.size(), equalTo(1));
     assertTrue(licenseState.activeUpdates.get(0));
  }

  public void testNoNotificationOnExistingLicense() throws Exception {
     final License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(24));
     MetaData metaData = MetaData.builder().putCustom(LicensesMetaData.TYPE, new
LicensesMetaData(license, null)).build();
     ClusterState newState = ClusterState.builder(new ClusterName("a")).metaData(metaData).build();
     ClusterState oldState = ClusterState.builder(newState).build();
     licenseService.clusterChanged(new ClusterChangedEvent("simulated", newState, oldState));
     assertThat(licenseState.activeUpdates.size(), equalTo(0));
  }

  public void testSelfGeneratedLicenseGeneration() throws Exception {
     DiscoveryNode master = new DiscoveryNode("b", buildNewFakeTransportAddress(), emptyMap(), emptySet(),
Version.CURRENT);
     ClusterState oldState = ClusterState.builder(new ClusterName("a"))
          .nodes(DiscoveryNodes.builder().masterNodeId(master.getId()).add(master)).build();
     when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
     ClusterState newState = ClusterState.builder(oldState).nodes(discoveryNodes).build();

     licenseService.clusterChanged(new ClusterChangedEvent("simulated", newState, oldState));
```

```
    ArgumentCaptor<ClusterStateUpdateTask> stateUpdater =
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
    verify(clusterService, times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
    ClusterState stateWithLicense = stateUpdater.getValue().execute(newState);
    LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
    assertNotNull(licenseMetaData);
    assertNotNull(licenseMetaData.getLicense());
    assertEquals(licenseType, licenseMetaData.getLicense().type());
    long expiration;
    if (licenseType.equals("basic")) {
      expiration = LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
    } else {
      expiration = LicenseService.NON_BASIC_SELF_GENERATED_LICENSE_DURATION.millis() +
clock.millis();
    }
    assertEquals(expiration, licenseMetaData.getLicense().expiryDate());
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.XPackSettings;

import java.util.Arrays;
import java.util.function.Predicate;
import java.util.stream.Collectors;

import static org.elasticsearch.license.License.OperationMode.BASIC;
import static org.elasticsearch.license.License.OperationMode.GOLD;
import static org.elasticsearch.license.License.OperationMode.MISSING;
import static org.elasticsearch.license.License.OperationMode.PLATINUM;
import static org.elasticsearch.license.License.OperationMode.STANDARD;
import static org.elasticsearch.license.License.OperationMode.TRIAL;
import static org.hamcrest.Matchers.is;

/**
* Unit tests for the {@link XPackLicenseState}
*/
public class XPackLicenseStateTests extends ESTestCase {
   /** Creates a license state with the given license type and active state, and checks the given method returns
expected. */
   void assertAllowed(OperationMode mode, boolean active, Predicate<XPackLicenseState> predicate, boolean
expected) {
       XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
       licenseState.update(mode, active);
       assertEquals(expected, predicate.test(licenseState));
   }

   /**
    * Checks the ack message going from the  {@code from} license type to {@code to} license type.
    * TODO: check the actual messages, not just the number of them! This was copied from previous license tests...
    */
   void assertAckMesssages(String feature, OperationMode from, OperationMode to, int expectedMessages) {
       String[] gotMessages = XPackLicenseState.ACKNOWLEDGMENT_MESSAGES.get(feature).apply(from, to);
       assertEquals(expectedMessages, gotMessages.length);
   }

   static <T> T randomFrom(T[] values, Predicate<T> filter) {
       return randomFrom(Arrays.stream(values).filter(filter).collect(Collectors.toList()));
   }

   static OperationMode randomMode() {
       return randomFrom(OperationMode.values());
   }

   public static OperationMode randomTrialStandardGoldOrPlatinumMode() {
       return randomFrom(TRIAL, STANDARD, GOLD, PLATINUM);
   }

   public static OperationMode randomTrialOrPlatinumMode() {
       return randomFrom(TRIAL, PLATINUM);
   }

   public static OperationMode randomTrialBasicStandardGoldOrPlatinumMode() {
       return randomFrom(TRIAL, BASIC, STANDARD, GOLD, PLATINUM);
   }

   public static OperationMode randomBasicStandardOrGold() {
       return randomFrom(BASIC, STANDARD, GOLD);
   }

   public void testSecurityDefaults() {
       XPackLicenseState licenseState =
               new XPackLicenseState(Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(),
true).build());
       assertThat(licenseState.isAuthAllowed(), is(true));
       assertThat(licenseState.isIpFilteringAllowed(), is(true));
```

```
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(true));

        licenseState =
            new XPackLicenseState(Settings.builder().put(XPackSettings.TRANSPORT_SSL_ENABLED.getKey(),
true).build());
        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(true));

        licenseState = new XPackLicenseState(Settings.EMPTY);
        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(true));
    }

    public void testSecurityBasic() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(BASIC, true);

        assertThat(licenseState.isAuthAllowed(), is(false));
        assertThat(licenseState.isIpFilteringAllowed(), is(false));
        assertThat(licenseState.isAuditingAllowed(), is(false));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.NONE));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityBasicExpired() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(BASIC, false);

        assertThat(licenseState.isAuthAllowed(), is(false));
        assertThat(licenseState.isIpFilteringAllowed(), is(false));
        assertThat(licenseState.isAuditingAllowed(), is(false));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(false));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.NONE));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityStandard() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(STANDARD, true);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(false));
        assertThat(licenseState.isAuditingAllowed(), is(false));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.NATIVE));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityStandardExpired() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(STANDARD, false);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(false));
        assertThat(licenseState.isAuditingAllowed(), is(false));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(false));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.NATIVE));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityGold() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(GOLD, true);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.DEFAULT));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }
```

```
    public void testSecurityGoldExpired() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(GOLD, false);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(false));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.DEFAULT));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityPlatinum() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(PLATINUM, true);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(true));
    }

    public void testSecurityPlatinumExpired() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(PLATINUM, false);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(false));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityAckBasicToNotGoldOrStandard() {
        OperationMode toMode = randomFrom(OperationMode.values(), mode -> mode != GOLD && mode !=
STANDARD);
        assertAckMesssages(XPackField.SECURITY, BASIC, toMode, 0);
    }
    public void testSecurityAckAnyToTrialOrPlatinum() {
        assertAckMesssages(XPackField.SECURITY, randomMode(), randomTrialOrPlatinumMode(), 0);
    }

    public void testSecurityAckTrialStandardGoldOrPlatinumToBasic() {
        assertAckMesssages(XPackField.SECURITY, randomTrialStandardGoldOrPlatinumMode(), BASIC, 3);
    }

    public void testSecurityAckAnyToStandard() {
        OperationMode from = randomFrom(BASIC, GOLD, PLATINUM, TRIAL);
        assertAckMesssages(XPackField.SECURITY, from, STANDARD, 4);
    }

    public void testSecurityAckBasicStandardTrialOrPlatinumToGold() {
        OperationMode from = randomFrom(BASIC, PLATINUM, TRIAL, STANDARD);
        assertAckMesssages(XPackField.SECURITY, from, GOLD, 2);
    }

    public void testMonitoringAckBasicToAny() {
        assertAckMesssages(XPackField.MONITORING, BASIC, randomMode(), 0);
    }

    public void testMonitoringAckAnyToTrialGoldOrPlatinum() {
        assertAckMesssages(XPackField.MONITORING, randomMode(),
randomTrialStandardGoldOrPlatinumMode(), 0);
    }

    public void testMonitoringAckNotBasicToBasic() {
        OperationMode from = randomFrom(STANDARD, GOLD, PLATINUM, TRIAL);
        assertAckMesssages(XPackField.MONITORING, from, BASIC, 2);
    }

    public void testMonitoringAllowed() {
        assertAllowed(randomMode(), true, XPackLicenseState::isMonitoringAllowed, true);
        assertAllowed(randomMode(), false, XPackLicenseState::isMonitoringAllowed, false);
    }

    public void testMonitoringUpdateRetention() {
        assertAllowed(STANDARD, true, XPackLicenseState::isUpdateRetentionAllowed, true);
        assertAllowed(GOLD, true, XPackLicenseState::isUpdateRetentionAllowed, true);
        assertAllowed(PLATINUM, true, XPackLicenseState::isUpdateRetentionAllowed, true);
        assertAllowed(TRIAL, true, XPackLicenseState::isUpdateRetentionAllowed, true);
        assertAllowed(BASIC, true, XPackLicenseState::isUpdateRetentionAllowed, false);
        assertAllowed(MISSING, false, XPackLicenseState::isUpdateRetentionAllowed, false);
    }

    public void testWatcherPlatinumGoldTrialStandard() throws Exception {
```

```
    assertAllowed(TRIAL, true, XPackLicenseState::isWatcherAllowed, true);
    assertAllowed(GOLD, true, XPackLicenseState::isWatcherAllowed, true);
    assertAllowed(PLATINUM, true, XPackLicenseState::isWatcherAllowed, true);
    assertAllowed(STANDARD, true, XPackLicenseState::isWatcherAllowed, true);
}

public void testWatcherBasicLicense() throws Exception {
    assertAllowed(BASIC, true, XPackLicenseState::isWatcherAllowed, false);
}

public void testWatcherInactive() {
    assertAllowed(BASIC, false, XPackLicenseState::isWatcherAllowed, false);
}

public void testWatcherInactivePlatinumGoldTrial() throws Exception {
    assertAllowed(TRIAL, false, XPackLicenseState::isWatcherAllowed, false);
    assertAllowed(GOLD, false, XPackLicenseState::isWatcherAllowed, false);
    assertAllowed(PLATINUM, false, XPackLicenseState::isWatcherAllowed, false);
    assertAllowed(STANDARD, false, XPackLicenseState::isWatcherAllowed, false);
}

public void testGraphPlatinumTrial() throws Exception {
    assertAllowed(TRIAL, true, XPackLicenseState::isGraphAllowed, true);
    assertAllowed(PLATINUM, true, XPackLicenseState::isGraphAllowed, true);
}

public void testGraphBasic() throws Exception {
    assertAllowed(BASIC, true, XPackLicenseState::isGraphAllowed, false);
}

public void testGraphStandard() throws Exception {
    assertAllowed(STANDARD, true, XPackLicenseState::isGraphAllowed, false);
}

public void testGraphInactiveBasic() {
    assertAllowed(BASIC, false, XPackLicenseState::isGraphAllowed, false);
}

public void testGraphInactivePlatinumTrial() throws Exception {
    assertAllowed(TRIAL, false, XPackLicenseState::isMachineLearningAllowed, false);
    assertAllowed(PLATINUM, false, XPackLicenseState::isMachineLearningAllowed, false);
}

public void testMachineLearningPlatinumTrial() throws Exception {
    assertAllowed(TRIAL, true, XPackLicenseState::isMachineLearningAllowed, true);
    assertAllowed(PLATINUM, true, XPackLicenseState::isMachineLearningAllowed, true);
}
```

```
public void testMachineLearningBasic() throws Exception {
    assertAllowed(BASIC, true, XPackLicenseState::isMachineLearningAllowed, false);
}

public void testMachineLearningStandard() throws Exception {
    assertAllowed(STANDARD, true, XPackLicenseState::isMachineLearningAllowed, false);
}

public void testMachineLearningInactiveBasic() {
    assertAllowed(BASIC, false, XPackLicenseState::isMachineLearningAllowed, false);
}

public void testMachineLearningInactivePlatinumTrial() throws Exception {
    assertAllowed(TRIAL, false, XPackLicenseState::isMachineLearningAllowed, false);
    assertAllowed(PLATINUM, false, XPackLicenseState::isMachineLearningAllowed, false);
}

public void testLogstashPlatinumGoldTrialStandard() throws Exception {
    assertAllowed(TRIAL, true, XPackLicenseState::isLogstashAllowed, true);
    assertAllowed(GOLD, true, XPackLicenseState::isLogstashAllowed, true);
    assertAllowed(PLATINUM, true, XPackLicenseState::isLogstashAllowed, true);
    assertAllowed(STANDARD, true, XPackLicenseState::isLogstashAllowed, true);
}

public void testLogstashBasicLicense() throws Exception {
    assertAllowed(BASIC, true, XPackLicenseState::isLogstashAllowed, false);
}

public void testLogstashInactive() {
    assertAllowed(BASIC, false, XPackLicenseState::isLogstashAllowed, false);
    assertAllowed(TRIAL, false, XPackLicenseState::isLogstashAllowed, false);
    assertAllowed(GOLD, false, XPackLicenseState::isLogstashAllowed, false);
    assertAllowed(PLATINUM, false, XPackLicenseState::isLogstashAllowed, false);
    assertAllowed(STANDARD, false, XPackLicenseState::isLogstashAllowed, false);
}

public void testSqlDefaults() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(true));
}

public void testSqlBasic() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(BASIC, true);

    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(false));
```

```
}

public void testSqlBasicExpired() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(BASIC, false);

    assertThat(licenseState.isSqlAllowed(), is(false));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlStandard() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(STANDARD, true);

    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlStandardExpired() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(STANDARD, false);

    assertThat(licenseState.isSqlAllowed(), is(false));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlGold() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(GOLD, true);

    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlGoldExpired() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(GOLD, false);

    assertThat(licenseState.isSqlAllowed(), is(false));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlPlatinum() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(PLATINUM, true);

    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(true));
```

```
  }

  public void testSqlPlatinumExpired() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(PLATINUM, false);

    assertThat(licenseState.isSqlAllowed(), is(false));
    assertThat(licenseState.isJdbcAllowed(), is(false));
  }

  public void testSqlAckAnyToTrialOrPlatinum() {
    assertAckMesssages(XPackField.SQL, randomMode(), randomTrialOrPlatinumMode(), 0);
  }

  public void testSqlAckTrialOrPlatinumToNotTrialOrPlatinum() {
    assertAckMesssages(XPackField.SQL, randomTrialOrPlatinumMode(), randomBasicStandardOrGold(), 1);
  }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.client.Response;
import org.elasticsearch.client.ResponseException;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.common.io.Streams;
import org.elasticsearch.common.network.NetworkModule;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackClientPlugin;

import java.io.InputStreamReader;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.Collection;

import static org.elasticsearch.test.ESIntegTestCase.Scope.SUITE;

@ESIntegTestCase.ClusterScope(scope = SUITE)
public class StartTrialLicenseTests extends AbstractLicensesIntegrationTestCase {
  @Override
  protected Settings nodeSettings(int nodeOrdinal) {
    return Settings.builder()
            .put(super.nodeSettings(nodeOrdinal))
            .put("node.data", true)
            .put(LicenseService.SELF_GENERATED_LICENSE_TYPE.getKey(), "basic")
            .put(NetworkModule.HTTP_ENABLED.getKey(), true).build();
  }

  @Override
  protected Collection<Class<? extends Plugin>> nodePlugins() {
    return Arrays.asList(LocalStateCompositeXPackPlugin.class, Netty4Plugin.class);
  }

  @Override
  protected Collection<Class<? extends Plugin>> transportClientPlugins() {
    return Arrays.asList(XPackClientPlugin.class, Netty4Plugin.class);
  }

  public void testStartTrial() throws Exception {
    LicensingClient licensingClient = new LicensingClient(client());
    ensureStartingWithBasic();

    RestClient restClient = getRestClient();
    Response response = restClient.performRequest("GET", "/_xpack/license/trial_status");
    String body = Streams.copyToString(new InputStreamReader(response.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response.getStatusLine().getStatusCode());
    assertEquals("{\"eligible_to_start_trial\":true}", body);

    // Test that starting will fail without acknowledgement
    Response response2 = restClient.performRequest("POST", "/_xpack/license/start_trial");
    String body2 = Streams.copyToString(new InputStreamReader(response2.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response2.getStatusLine().getStatusCode());
    assertTrue(body2.contains("\"trial_was_started\":false"));
    assertTrue(body2.contains("\"error_message\":\"Operation failed: Needs acknowledgement.\""));
    assertTrue(body2.contains("\"acknowledged\":false"));

    assertBusy(() -> {
        GetLicenseResponse getLicenseResponse = licensingClient.prepareGetLicense().get();
        assertEquals("basic", getLicenseResponse.license().type());
    });

    String type = randomFrom(LicenseService.VALID_TRIAL_TYPES);

    Response response3 = restClient.performRequest("POST",
"/_xpack/license/start_trial?acknowledge=true&type=" + type);
```

```
    String body3 = Streams.copyToString(new InputStreamReader(response3.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response3.getStatusLine().getStatusCode());
    assertTrue(body3.contains("\"trial_was_started\":true"));
    assertTrue(body3.contains("\"type\":\"" + type + "\""));
    assertTrue(body3.contains("\"acknowledged\":true"));

    assertBusy(() -> {
      GetLicenseResponse postTrialLicenseResponse = licensingClient.prepareGetLicense().get();
      assertEquals(type, postTrialLicenseResponse.license().type());
    });

    Response response4 = restClient.performRequest("GET", "/_xpack/license/trial_status");
    String body4 = Streams.copyToString(new InputStreamReader(response4.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response4.getStatusLine().getStatusCode());
    assertEquals("{\"eligible_to_start_trial\":false}", body4);

    String secondAttemptType = randomFrom(LicenseService.VALID_TRIAL_TYPES);

    ResponseException ex = expectThrows(ResponseException.class,
        () -> restClient.performRequest("POST", "/_xpack/license/start_trial?acknowledge=true&type=" +
secondAttemptType));
    Response response5 = ex.getResponse();
    String body5 = Streams.copyToString(new InputStreamReader(response5.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(403, response5.getStatusLine().getStatusCode());
    assertTrue(body5.contains("\"trial_was_started\":false"));
    assertTrue(body5.contains("\"error_message\":\"Operation failed: Trial was already activated.\""));
  }

  public void testInvalidType() throws Exception {
    ensureStartingWithBasic();

    ResponseException ex = expectThrows(ResponseException.class, () ->
        getRestClient().performRequest("POST", "/_xpack/license/start_trial?type=basic"));
    Response response = ex.getResponse();
    String body = Streams.copyToString(new InputStreamReader(response.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(400, response.getStatusLine().getStatusCode());
    assertTrue(body.contains("\"type\":\"illegal_argument_exception\""));
    assertTrue(body.contains("\"reason\":\"Cannot start trial of type [basic]. Valid trial types are ["));
  }

  private void ensureStartingWithBasic() throws Exception {
    LicensingClient licensingClient = new LicensingClient(client());
    GetLicenseResponse getLicenseResponse = licensingClient.prepareGetLicense().get();
```

```
        if ("basic".equals(getLicenseResponse.license().type()) == false) {
            licensingClient.preparePostStartBasic().setAcknowledge(true).get();
        }

        assertBusy(() -> {
            GetLicenseResponse postTrialLicenseResponse = licensingClient.prepareGetLicense().get();
            assertEquals("basic", postTrialLicenseResponse.license().type());
        });
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.analysis.common.CommonAnalysisPlugin;
import org.elasticsearch.common.network.NetworkModule;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.env.Environment;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase.ClusterScope;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.Arrays;
import java.util.Collection;

import static org.elasticsearch.test.ESIntegTestCase.Scope.TEST;
import static org.hamcrest.CoreMatchers.equalTo;
import static org.hamcrest.CoreMatchers.nullValue;

@ClusterScope(scope = TEST, numDataNodes = 0, numClientNodes = 0, maxNumDataNodes = 0,
transportClientRatio = 0)
public class LicenseServiceClusterTests extends AbstractLicensesIntegrationTestCase {

    @Override
    protected Settings nodeSettings(int nodeOrdinal) {
        return nodeSettingsBuilder(nodeOrdinal).build();
    }

    private Settings.Builder nodeSettingsBuilder(int nodeOrdinal) {
        return Settings.builder()
                .put(super.nodeSettings(nodeOrdinal))
                .put("node.data", true)
                .put("resource.reload.interval.high", "500ms") // for license mode file watcher
                .put(NetworkModule.HTTP_ENABLED.getKey(), true);
    }

    @Override
    protected Collection<Class<? extends Plugin>> nodePlugins() {
        return Arrays.asList(LocalStateCompositeXPackPlugin.class, CommonAnalysisPlugin.class,
Netty4Plugin.class);
    }

    @Override
    protected Collection<Class<? extends Plugin>> transportClientPlugins() {
        return nodePlugins();
    }

    public void testClusterRestartWithLicense() throws Exception {
        wipeAllLicenses();

        int numNodes = randomIntBetween(1, 5);
        logger.info("--> starting {} node(s)", numNodes);
        internalCluster().startNodes(numNodes);
        ensureGreen();

        logger.info("--> put signed license");
        LicensingClient licensingClient = new LicensingClient(client());
        License license = TestUtils.generateSignedLicense(TimeValue.timeValueMinutes(1));
        putLicense(license);
        assertThat(licensingClient.prepareGetLicense().get().license(), equalTo(license));
        assertOperationMode(license.operationMode());

        logger.info("--> restart all nodes");
        internalCluster().fullRestart();
        ensureYellow();
        licensingClient = new LicensingClient(client());
        logger.info("--> get and check signed license");
        assertThat(licensingClient.prepareGetLicense().get().license(), equalTo(license));
        logger.info("--> remove licenses");
        licensingClient.prepareDeleteLicense().get();
        assertOperationMode(License.OperationMode.MISSING);

        logger.info("--> restart all nodes");
        internalCluster().fullRestart();
        licensingClient = new LicensingClient(client());
        ensureYellow();
        assertThat(licensingClient.prepareGetLicense().get().license(), nullValue());
        assertOperationMode(License.OperationMode.MISSING);


        wipeAllLicenses();
    }

    public void testCloudInternalLicense() throws Exception {
        wipeAllLicenses();

        int numNodes = randomIntBetween(1, 5);
        logger.info("--> starting {} node(s)", numNodes);
        internalCluster().startNodes(numNodes);
        ensureGreen();

        logger.info("--> put signed license");
        LicensingClient licensingClient = new LicensingClient(client());
        License license = TestUtils.generateSignedLicense("cloud_internal", License.VERSION_CURRENT,
System.currentTimeMillis(),
                TimeValue.timeValueMinutes(1));
        putLicense(license);
        assertThat(licensingClient.prepareGetLicense().get().license(), equalTo(license));
        assertOperationMode(License.OperationMode.PLATINUM);
        writeCloudInternalMode("gold");
        assertOperationMode(License.OperationMode.GOLD);
        writeCloudInternalMode("basic");
        assertOperationMode(License.OperationMode.BASIC);
    }

    public void testClusterRestartWhileEnabled() throws Exception {
        wipeAllLicenses();
        internalCluster().startNode();
        ensureGreen();
        assertLicenseActive(true);
        logger.info("--> restart node");
        internalCluster().fullRestart();
        ensureYellow();
        logger.info("--> await node for enabled");
        assertLicenseActive(true);
    }

    public void testClusterRestartWhileGrace() throws Exception {
        wipeAllLicenses();
        internalCluster().startNode();
        assertLicenseActive(true);
        putLicense(TestUtils.generateSignedLicense(TimeValue.timeValueMillis(0)));
        ensureGreen();
        assertLicenseActive(true);
        logger.info("--> restart node");
        internalCluster().fullRestart();
        ensureYellow();
        logger.info("--> await node for grace_period");
        assertLicenseActive(true);
    }

    public void testClusterRestartWhileExpired() throws Exception {
        wipeAllLicenses();
        internalCluster().startNode();
        ensureGreen();
        assertLicenseActive(true);
        putLicense(TestUtils.generateExpiredNonBasicLicense(System.currentTimeMillis() -
LicenseService.GRACE_PERIOD_DURATION.getMillis()));
        assertLicenseActive(false);
        logger.info("--> restart node");
        internalCluster().fullRestart();
        ensureYellow();
        logger.info("--> await node for disabled");
        assertLicenseActive(false);
    }

    private void assertOperationMode(License.OperationMode operationMode) throws InterruptedException {
        boolean success = awaitBusy(() -> {
            for (XPackLicenseState licenseState : internalCluster().getDataNodeInstances(XPackLicenseState.class)) {
                if (licenseState.getOperationMode() == operationMode) {
                    return true;
                }
            }
            return false;
        });
        assertTrue(success);
    }

    private void writeCloudInternalMode(String mode) throws Exception {
        for (Environment environment : internalCluster().getDataOrMasterNodeInstances(Environment.class)) {
            Path licenseModePath = XPackPlugin.resolveConfigFile(environment, "license_mode");
            Files.createDirectories(licenseModePath.getParent());
            Files.write(licenseModePath, mode.getBytes(StandardCharsets.UTF_8));
        }
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.test.ESTestCase;

import java.util.Arrays;

import static org.hamcrest.Matchers.contains;
import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.hasSize;

public class LicenseUtilsTests extends ESTestCase {

  public void testNewExpirationException() {
    for (String feature : Arrays.asList("feature", randomAlphaOfLength(5), null, "")) {
      ElasticsearchSecurityException exception = LicenseUtils.newComplianceException(feature);
      assertNotNull(exception);
      assertThat(exception.getMetadataKeys(), contains(LicenseUtils.EXPIRED_FEATURE_METADATA));
      assertThat(exception.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA), hasSize(1));
      assertThat(exception.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA).iterator().next(),
equalTo(feature));
    }
  }

  public void testIsLicenseExpiredException() {
    ElasticsearchSecurityException exception = LicenseUtils.newComplianceException("feature");
    assertTrue(LicenseUtils.isLicenseExpiredException(exception));

    exception = new ElasticsearchSecurityException("msg");
    assertFalse(LicenseUtils.isLicenseExpiredException(exception));
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.watcher.FileWatcher;
import org.elasticsearch.watcher.ResourceWatcherService;
import org.junit.Before;

import java.nio.file.Path;

import static org.hamcrest.Matchers.equalTo;
import static org.mockito.Matchers.any;
import static org.mockito.Matchers.eq;
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.times;
import static org.mockito.Mockito.verify;
import static org.mockito.Mockito.verifyZeroInteractions;

public class LicenseOperationModeUpdateTests extends ESTestCase {

  private OperationModeFileWatcher operationModeFileWatcher;
  private Path licenseModeFile;
  private ResourceWatcherService resourceWatcherService;

  @Before
  public void init() throws Exception {
     licenseModeFile = createTempFile();
     resourceWatcherService = mock(ResourceWatcherService.class);
     operationModeFileWatcher = new OperationModeFileWatcher(resourceWatcherService, licenseModeFile,
logger, () -> {});
  }

  public void testLicenseOperationModeUpdate() throws Exception {
     String type = randomFrom("trial", "basic", "standard", "gold", "platinum");
     License license = License.builder()
          .uid("id")
          .expiryDate(0)
          .issueDate(0)
          .issuedTo("elasticsearch")
          .issuer("issuer")
          .type(type)
          .maxNodes(1)
          .build();

     assertThat(license.operationMode(), equalTo(License.OperationMode.resolve(type)));
     OperationModeFileWatcherTests.writeMode("gold", licenseModeFile);
     license.setOperationModeFileWatcher(operationModeFileWatcher);
     verifyZeroInteractions(resourceWatcherService);
     assertThat(license.operationMode(), equalTo(License.OperationMode.resolve(type)));
  }

  public void testCloudInternalLicenseOperationModeUpdate() throws Exception {
     License license = License.builder()
          .uid("id")
          .expiryDate(0)
          .issueDate(0)
          .issuedTo("elasticsearch")
          .issuer("issuer")
          .type("cloud_internal")
          .maxNodes(1)
          .build();
```

```
        assertThat(license.operationMode(), equalTo(License.OperationMode.PLATINUM));
        OperationModeFileWatcherTests.writeMode("gold", licenseModeFile);
        license.setOperationModeFileWatcher(operationModeFileWatcher);
        verify(resourceWatcherService, times(1)).add(any(FileWatcher.class),
eq(ResourceWatcherService.Frequency.HIGH));
        assertThat(license.operationMode(), equalTo(License.OperationMode.GOLD));
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.xpack.core.scheduler.SchedulerEngine;
import org.junit.Before;

import static org.hamcrest.Matchers.equalTo;

public class LicenseScheduleTests extends ESTestCase {

    private License license;
    private SchedulerEngine.Schedule schedule;

    @Before
    public void setuo() throws Exception {
        license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(12));
        schedule = LicenseService.nextLicenseCheck(license);
    }

    public void testEnabledLicenseSchedule() throws Exception {
        int expiryDuration = (int) (license.expiryDate() - license.issueDate());
        long triggeredTime = license.issueDate() + between(0, expiryDuration);
        assertThat(schedule.nextScheduledTimeAfter(license.issueDate(), triggeredTime),
equalTo(license.expiryDate()));
    }

    public void testGraceLicenseSchedule() throws Exception {
        long triggeredTime = license.expiryDate() + between(1,
                ((int) LicenseService.GRACE_PERIOD_DURATION.getMillis()));
        assertThat(schedule.nextScheduledTimeAfter(license.issueDate(), triggeredTime),
                equalTo(license.expiryDate() + LicenseService.GRACE_PERIOD_DURATION.getMillis()));
    }
```

```
  public void testExpiredLicenseSchedule() throws Exception {
    long triggeredTime = license.expiryDate() + LicenseService.GRACE_PERIOD_DURATION.getMillis() +
        randomIntBetween(1, 1000);
    assertThat(schedule.nextScheduledTimeAfter(license.issueDate(), triggeredTime),
        equalTo(-1L));
  }

  public void testInvalidLicenseSchedule() throws Exception {
    long triggeredTime = license.issueDate() - randomIntBetween(1, 1000);
    assertThat(schedule.nextScheduledTimeAfter(triggeredTime, triggeredTime),
        equalTo(license.issueDate()));
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.test.ESTestCase;

import static org.elasticsearch.common.unit.TimeValue.timeValueMillis;
import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.nullValue;

public class ExpirationCallbackTests extends ESTestCase {

  public void testPostExpirationDelay() throws Exception {
    TimeValue expiryDuration = TimeValue.timeValueSeconds(randomIntBetween(5, 10));
    TimeValue min = TimeValue.timeValueSeconds(1);
    TimeValue max = TimeValue.timeValueSeconds(4);
    TimeValue frequency = TimeValue.timeValueSeconds(1);
    NoopPostExpirationCallback post = new NoopPostExpirationCallback(min, max, frequency);
    long now = System.currentTimeMillis();
    long expiryDate = now + expiryDuration.getMillis();
    assertThat(post.delay(expiryDate, now),
        equalTo(TimeValue.timeValueMillis(expiryDuration.getMillis() + min.getMillis()))); // before license
expiry
    assertThat(post.delay(expiryDate, expiryDate), equalTo(min)); // on license expiry
    int latestValidTriggerDelay = (int) (expiryDuration.getMillis() + max.getMillis());
    int earliestValidTriggerDelay = (int) (expiryDuration.getMillis() + min.getMillis());
    assertExpirationCallbackDelay(post, expiryDuration.millis(), latestValidTriggerDelay,
earliestValidTriggerDelay);
  }

  public void testPreExpirationDelay() throws Exception {
    TimeValue expiryDuration = TimeValue.timeValueSeconds(randomIntBetween(5, 10));
    TimeValue min = TimeValue.timeValueSeconds(1);
    TimeValue max = TimeValue.timeValueSeconds(4);
    TimeValue frequency = TimeValue.timeValueSeconds(1);
    NoopPreExpirationCallback pre = new NoopPreExpirationCallback(min, max, frequency);
    long now = System.currentTimeMillis();
    long expiryDate = now + expiryDuration.getMillis();
    assertThat(pre.delay(expiryDate, expiryDate), nullValue()); // on license expiry
    int latestValidTriggerDelay = (int) (expiryDuration.getMillis() - min.getMillis());
    int earliestValidTriggerDelay = (int) (expiryDuration.getMillis() - max.getMillis());
    assertExpirationCallbackDelay(pre, expiryDuration.millis(), latestValidTriggerDelay,
earliestValidTriggerDelay);
  }

  public void testPostExpirationWithNullMax() throws Exception {
    int postExpirySeconds = randomIntBetween(5, 10);
    TimeValue postExpiryDuration = TimeValue.timeValueSeconds(postExpirySeconds);
    TimeValue min = TimeValue.timeValueSeconds(postExpirySeconds - randomIntBetween(1, 3));

    final ExpirationCallback.Post post = new NoopPostExpirationCallback(min, null, timeValueMillis(10));
    long now = System.currentTimeMillis();
    assertThat(post.delay(now - postExpiryDuration.millis(), now), equalTo(TimeValue.timeValueMillis(0)));
  }

  public void testPreExpirationWithNullMin() throws Exception {
    int expirySeconds = randomIntBetween(5, 10);
    TimeValue expiryDuration = TimeValue.timeValueSeconds(expirySeconds);
    TimeValue max = TimeValue.timeValueSeconds(expirySeconds + randomIntBetween(1, 10));

    final ExpirationCallback.Pre pre = new NoopPreExpirationCallback(null, max, timeValueMillis(10));
    long now = System.currentTimeMillis();
    assertThat(pre.delay(expiryDuration.millis() + now, now), equalTo(TimeValue.timeValueMillis(0)));
  }

  public void testPreExpirationScheduleTime() throws Exception {
    TimeValue expiryDuration = TimeValue.timeValueSeconds(randomIntBetween(5, 10));
    TimeValue min = TimeValue.timeValueSeconds(1);
    TimeValue max = TimeValue.timeValueSeconds(4);
    TimeValue frequency = TimeValue.timeValueSeconds(1);
    NoopPreExpirationCallback pre = new NoopPreExpirationCallback(min, max, frequency);
    int latestValidTriggerDelay = (int) (expiryDuration.getMillis() - min.getMillis());
    int earliestValidTriggerDelay = (int) (expiryDuration.getMillis() - max.getMillis());
    assertExpirationCallbackScheduleTime(pre, expiryDuration.millis(), latestValidTriggerDelay,
earliestValidTriggerDelay);
  }

  public void testPostExpirationScheduleTime() throws Exception {
    TimeValue expiryDuration = TimeValue.timeValueSeconds(randomIntBetween(5, 10));
    TimeValue min = TimeValue.timeValueSeconds(1);
    TimeValue max = TimeValue.timeValueSeconds(4);
    TimeValue frequency = TimeValue.timeValueSeconds(1);
    NoopPostExpirationCallback pre = new NoopPostExpirationCallback(min, max, frequency);
    int latestValidTriggerDelay = (int) (expiryDuration.getMillis() + max.getMillis());
    int earliestValidTriggerDelay = (int) (expiryDuration.getMillis() + min.getMillis());
    assertExpirationCallbackScheduleTime(pre, expiryDuration.millis(), latestValidTriggerDelay,
earliestValidTriggerDelay);
  }

  private void assertExpirationCallbackDelay(ExpirationCallback expirationCallback, long expiryDuration,
                        int latestValidTriggerDelay, int earliestValidTriggerDelay) {
    long now = System.currentTimeMillis();
    long expiryDate = now + expiryDuration;
    // bounds
    assertThat(expirationCallback.delay(expiryDate, now + earliestValidTriggerDelay),
equalTo(TimeValue.timeValueMillis(0)));
    assertThat(expirationCallback.delay(expiryDate, now + latestValidTriggerDelay),
equalTo(TimeValue.timeValueMillis(0)));
    // in match
    assertThat(expirationCallback.delay(expiryDate,
        now + randomIntBetween(earliestValidTriggerDelay, latestValidTriggerDelay)),
        equalTo(TimeValue.timeValueMillis(0)));
    // out of bounds
    int deltaBeforeEarliestMatch = between(1, earliestValidTriggerDelay);
    assertThat(expirationCallback.delay(expiryDate, now + deltaBeforeEarliestMatch),
        equalTo(TimeValue.timeValueMillis(earliestValidTriggerDelay - deltaBeforeEarliestMatch)));
    int deltaAfterLatestMatch = between(latestValidTriggerDelay + 1, Integer.MAX_VALUE); // after expiry and
after max
    assertThat(expirationCallback.delay(expiryDate, expiryDate + deltaAfterLatestMatch), nullValue());
  }

  public void assertExpirationCallbackScheduleTime(ExpirationCallback expirationCallback, long expiryDuration,
                            int latestValidTriggerDelay, int earliestValidTriggerDelay) {
    long now = System.currentTimeMillis();
    long expiryDate = now + expiryDuration;
    int validTriggerInterval = between(earliestValidTriggerDelay, latestValidTriggerDelay);
    assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate,
        now + validTriggerInterval, now + validTriggerInterval),
        equalTo(now + validTriggerInterval));
    assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate, now, now + validTriggerInterval),
        equalTo(now + validTriggerInterval + expirationCallback.getFrequency()));

    int deltaBeforeEarliestMatch = between(1, earliestValidTriggerDelay - 1);
    assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate, now, now +
deltaBeforeEarliestMatch),
        equalTo(now + deltaBeforeEarliestMatch +
            expirationCallback.delay(expiryDate, now + deltaBeforeEarliestMatch).getMillis()));
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate,
                now + deltaBeforeEarliestMatch, now + deltaBeforeEarliestMatch),
                equalTo(now + deltaBeforeEarliestMatch +
                        expirationCallback.delay(expiryDate, now + deltaBeforeEarliestMatch).getMillis()));

        int deltaAfterLatestMatch = between(latestValidTriggerDelay + 1, Integer.MAX_VALUE); // after expiry and
after max
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate, now, now + deltaAfterLatestMatch),
equalTo(-1L));
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate,
                now + deltaAfterLatestMatch, now + deltaAfterLatestMatch),
                equalTo(-1L));
    }

    private static class NoopPostExpirationCallback extends ExpirationCallback.Post {

        NoopPostExpirationCallback(TimeValue min, TimeValue max, TimeValue frequency) {
            super(min, max, frequency);
        }

        @Override
        public void on(License license) {}
    }

    private static class NoopPreExpirationCallback extends ExpirationCallback.Pre {

        NoopPreExpirationCallback(TimeValue min, TimeValue max, TimeValue frequency) {
            super(min, max, frequency);
        }

        @Override
        public void on(License license) {}
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import com.carrotsearch.randomizedtesting.RandomizedTest;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.joda.DateMathParser;
import org.elasticsearch.common.joda.FormatDateTimeFormatter;
import org.elasticsearch.common.joda.Joda;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.licensor.LicenseSigner;
import org.hamcrest.MatcherAssert;
import org.joda.time.format.DateTimeFormatter;
import org.junit.Assert;

import java.io.IOException;
import java.io.InputStream;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.StandardCopyOption;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.atomic.AtomicReference;

import static com.carrotsearch.randomizedtesting.RandomizedTest.randomBoolean;
import static com.carrotsearch.randomizedtesting.RandomizedTest.randomInt;
import static org.apache.lucene.util.LuceneTestCase.createTempFile;
import static org.elasticsearch.common.xcontent.XContentFactory.jsonBuilder;
import static org.elasticsearch.test.ESTestCase.randomAlphaOfLength;
import static org.elasticsearch.test.ESTestCase.randomFrom;
import static org.elasticsearch.test.ESTestCase.randomIntBetween;
import static org.hamcrest.core.IsEqual.equalTo;
import static org.junit.Assert.assertThat;

public class TestUtils {

  private static final FormatDateTimeFormatter formatDateTimeFormatter = Joda.forPattern("yyyy-MM-dd");
  private static final DateMathParser dateMathParser = new DateMathParser(formatDateTimeFormatter);
  private static final DateTimeFormatter dateTimeFormatter = formatDateTimeFormatter.printer();

  public static String dateMathString(String time, final long now) {
    return dateTimeFormatter.print(dateMathParser.parse(time, () -> now));
  }

  public static long dateMath(String time, final long now) {
    return dateMathParser.parse(time, () -> now);
  }

  public static LicenseSpec generateRandomLicenseSpec(int version) {
    boolean datesInMillis = randomBoolean();
```

```
        long now = System.currentTimeMillis();
        String uid = UUID.randomUUID().toString();
        String feature = "feature__" + randomInt();
        String issuer = "issuer__"  + randomInt();
        String issuedTo = "issuedTo__" + randomInt();
        final String type;
        final String subscriptionType;
        if (version < License.VERSION_NO_FEATURE_TYPE) {
            subscriptionType = randomFrom("gold", "silver", "platinum");
            type = "subscription";//randomFrom("subscription", "internal", "development");
        } else {
            subscriptionType = null;
            type = randomFrom("basic", "dev", "gold", "silver", "platinum");
        }
        int maxNodes = RandomizedTest.randomIntBetween(5, 100);
        if (datesInMillis) {
            long issueDateInMillis = dateMath("now", now);
            long expiryDateInMillis = dateMath("now+10d/d", now);
            return new LicenseSpec(version, uid, feature, issueDateInMillis, expiryDateInMillis, type, subscriptionType,
issuedTo, issuer,
                maxNodes);
        } else {
            String issueDate = dateMathString("now", now);
            String expiryDate = dateMathString("now+10d/d", now);
            return new LicenseSpec(version, uid, feature, issueDate, expiryDate, type, subscriptionType, issuedTo,
issuer, maxNodes);
        }
    }

    public static String generateLicenseSpecString(LicenseSpec licenseSpec) throws IOException {
        XContentBuilder licenses = jsonBuilder();
        licenses.startObject();
        licenses.startArray("licenses");
        licenses.startObject()
            .field("uid", licenseSpec.uid)
            .field("type", licenseSpec.type)
            .field("subscription_type", licenseSpec.subscriptionType)
            .field("issued_to", licenseSpec.issuedTo)
            .field("issuer", licenseSpec.issuer)
            .field("feature", licenseSpec.feature)
            .field("max_nodes", licenseSpec.maxNodes);

        if (licenseSpec.issueDate != null) {
            licenses.field("issue_date", licenseSpec.issueDate);
        } else {
            licenses.field("issue_date_in_millis", licenseSpec.issueDateInMillis);
        }
        if (licenseSpec.expiryDate != null) {
```

```
        licenses.field("expiry_date", licenseSpec.expiryDate);
    } else {
        licenses.field("expiry_date_in_millis", licenseSpec.expiryDateInMillis);
    }
    licenses.field("version", licenseSpec.version);
    licenses.endObject();
    licenses.endArray();
    licenses.endObject();
    return Strings.toString(licenses);
}

public static License generateLicenses(LicenseSpec spec) {
    License.Builder builder = License.builder()
        .uid(spec.uid)
        .feature(spec.feature)
        .type(spec.type)
        .subscriptionType(spec.subscriptionType)
        .issuedTo(spec.issuedTo)
        .issuer(spec.issuer)
        .maxNodes(spec.maxNodes);

    if (spec.expiryDate != null) {
        builder.expiryDate(DateUtils.endOfTheDay(spec.expiryDate));
    } else {
        builder.expiryDate(spec.expiryDateInMillis);
    }
    if (spec.issueDate != null) {
        builder.issueDate(DateUtils.beginningOfTheDay(spec.issueDate));
    } else {
        builder.issueDate(spec.issueDateInMillis);
    }
    return builder.build();
}

public static void assertLicenseSpec(LicenseSpec spec, License license) {
    MatcherAssert.assertThat(license.uid(), equalTo(spec.uid));
    MatcherAssert.assertThat(license.issuedTo(), equalTo(spec.issuedTo));
    MatcherAssert.assertThat(license.issuer(), equalTo(spec.issuer));
    MatcherAssert.assertThat(license.type(), equalTo(spec.type));
    MatcherAssert.assertThat(license.maxNodes(), equalTo(spec.maxNodes));
    if (spec.issueDate != null) {
        MatcherAssert.assertThat(license.issueDate(), equalTo(DateUtils.beginningOfTheDay(spec.issueDate)));
    } else {
        MatcherAssert.assertThat(license.issueDate(), equalTo(spec.issueDateInMillis));
    }
    if (spec.expiryDate != null) {
        MatcherAssert.assertThat(license.expiryDate(), equalTo(DateUtils.endOfTheDay(spec.expiryDate)));
    } else {
            MatcherAssert.assertThat(license.expiryDate(), equalTo(spec.expiryDateInMillis));
        }
    }

    public static class LicenseSpec {
        public final int version;
        public final String feature;
        public final String issueDate;
        public final long issueDateInMillis;
        public final String expiryDate;
        public final long expiryDateInMillis;
        public final String uid;
        public final String type;
        public final String subscriptionType;
        public final String issuedTo;
        public final String issuer;
        public final int maxNodes;

        public LicenseSpec(String issueDate, String expiryDate) {
            this(License.VERSION_CURRENT, UUID.randomUUID().toString(), "feature", issueDate, expiryDate,
"trial", "none", "customer",
                "elasticsearch", 5);
        }

        public LicenseSpec(int version, String uid, String feature, long issueDateInMillis, long expiryDateInMillis,
String type,
                           String subscriptionType, String issuedTo, String issuer, int maxNodes) {
            this.version = version;
            this.feature = feature;
            this.issueDateInMillis = issueDateInMillis;
            this.issueDate = null;
            this.expiryDateInMillis = expiryDateInMillis;
            this.expiryDate = null;
            this.uid = uid;
            this.type = type;
            this.subscriptionType = subscriptionType;
            this.issuedTo = issuedTo;
            this.issuer = issuer;
            this.maxNodes = maxNodes;
        }

        public LicenseSpec(int version, String uid, String feature, String issueDate, String expiryDate, String type,
                           String subscriptionType, String issuedTo, String issuer, int maxNodes) {
            this.version = version;
            this.feature = feature;
            this.issueDate = issueDate;
            this.issueDateInMillis = -1;
            this.expiryDate = expiryDate;
            this.expiryDateInMillis = -1;
            this.uid = uid;
            this.type = type;
            this.subscriptionType = subscriptionType;
            this.issuedTo = issuedTo;
            this.issuer = issuer;
            this.maxNodes = maxNodes;
        }
    }

    public static Path getTestPriKeyPath() throws Exception {
        return getResourcePath("/private.key");
    }

    public static Path getTestPubKeyPath() throws Exception {
        return getResourcePath("/public.key");
    }

    public static String dumpLicense(License license) throws Exception {
        XContentBuilder builder = XContentFactory.contentBuilder(XContentType.JSON);
        builder.startObject();
        builder.startObject("license");
        license.toInnerXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.endObject();
        builder.endObject();
        return Strings.toString(builder);
    }

    public static License generateSignedLicense(TimeValue expiryDuration) throws Exception {
        return generateSignedLicense(null, -1, expiryDuration);
    }

    public static License generateSignedLicense(String type, TimeValue expiryDuration) throws Exception {
        return generateSignedLicense(type, -1, expiryDuration);
    }

    public static License generateSignedLicense(long issueDate, TimeValue expiryDuration) throws Exception {
        return generateSignedLicense(null, issueDate, expiryDuration);
    }

    public static License generateSignedLicense(String type, long issueDate, TimeValue expiryDuration) throws
Exception {
        return generateSignedLicense(type, randomIntBetween(License.VERSION_START,
License.VERSION_CURRENT), issueDate, expiryDuration);
    }

    /**
     * This method which chooses the license type randomly if the type is null. However, it will not randomly
     * choose trial or basic types as those types can only be self-generated.
     */
    public static License generateSignedLicense(String type, int version, long issueDate, TimeValue expiryDuration)
throws Exception {
        long issue = (issueDate != -1L) ? issueDate : System.currentTimeMillis() -
TimeValue.timeValueHours(2).getMillis();
        final String licenseType;
        if (version < License.VERSION_NO_FEATURE_TYPE) {
            licenseType = randomFrom("subscription", "internal", "development");
        } else {
            licenseType = (type != null) ? type : randomFrom("silver", "dev", "gold", "platinum");
        }
        final License.Builder builder = License.builder()
                .uid(UUID.randomUUID().toString())
                .version(version)
                .expiryDate(System.currentTimeMillis() + expiryDuration.getMillis())
                .issueDate(issue)
                .type(licenseType)
                .issuedTo("customer")
                .issuer("elasticsearch")
                .maxNodes(5);
        if (version == License.VERSION_START) {
            builder.subscriptionType((type != null) ? type : randomFrom("dev", "gold", "platinum", "silver"));
            builder.feature(randomAlphaOfLength(10));
        }
        LicenseSigner signer = new LicenseSigner(getTestPriKeyPath(), getTestPubKeyPath());
        return signer.sign(builder.build());
    }

    public static License generateSignedLicense(License.Builder builder) throws Exception {
        LicenseSigner signer = new LicenseSigner(getTestPriKeyPath(), getTestPubKeyPath());
        return signer.sign(builder.build());
    }

    public static License generateExpiredNonBasicLicense(long expiryDate) throws Exception {
        return generateExpiredNonBasicLicense(randomFrom("silver", "dev", "gold", "platinum"), expiryDate);
    }

    public static License generateExpiredNonBasicLicense() throws Exception {
        return generateExpiredNonBasicLicense(randomFrom("silver", "dev", "gold", "platinum"));
    }

    public static License generateExpiredNonBasicLicense(String type) throws Exception {
        return generateExpiredNonBasicLicense(type,
                System.currentTimeMillis() - TimeValue.timeValueHours(randomIntBetween(1, 10)).getMillis());
    }

    public static License generateExpiredNonBasicLicense(String type, long expiryDate) throws Exception {
```

```
    final License.Builder builder = License.builder()
            .uid(UUID.randomUUID().toString())
            .version(License.VERSION_CURRENT)
            .expiryDate(expiryDate)
            .issueDate(expiryDate - TimeValue.timeValueMinutes(10).getMillis())
            .type(type)
            .issuedTo("customer")
            .issuer("elasticsearch")
            .maxNodes(5);
    LicenseSigner signer = new LicenseSigner(getTestPriKeyPath(), getTestPubKeyPath());
    return signer.sign(builder.build());
}

private static Path getResourcePath(String resource) throws Exception {
    Path resourceFile = createTempFile();
    try (InputStream resourceInput = TestUtils.class.getResourceAsStream(resource)) {
        Files.copy(resourceInput, resourceFile, StandardCopyOption.REPLACE_EXISTING);
    }
    return resourceFile;
}

public static void registerAndAckSignedLicenses(final LicenseService licenseService, License license,
                                                final LicensesStatus expectedStatus) {
    PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(license).acknowledge(true);
    final CountDownLatch latch = new CountDownLatch(1);
    final AtomicReference<LicensesStatus> status = new AtomicReference<>();
    licenseService.registerLicense(putLicenseRequest, new ActionListener<PutLicenseResponse>() {
        @Override
        public void onResponse(PutLicenseResponse licensesUpdateResponse) {
            status.set(licensesUpdateResponse.status());
            latch.countDown();
        }

        @Override
        public void onFailure(Exception e) {
            latch.countDown();
        }
    });
    try {
        latch.await();
    } catch (InterruptedException e) {
        Assert.fail(e.getMessage());
    }
    assertThat(status.get(), equalTo(expectedStatus));
}

public static class AssertingLicenseState extends XPackLicenseState {
    public final List<License.OperationMode> modeUpdates = new ArrayList<>();
    public final List<Boolean> activeUpdates = new ArrayList<>();

    public AssertingLicenseState() {
      super(Settings.EMPTY);
    }

    @Override
    void update(License.OperationMode mode, boolean active) {
      modeUpdates.add(mode);
      activeUpdates.add(active);
    }
  }

  /**
   * A license state that makes the {@link #update(License.OperationMode, boolean)}
   * method public for use in tests.
   */
  public static class UpdatableLicenseState extends XPackLicenseState {
    public UpdatableLicenseState() {
      this(Settings.EMPTY);
    }

    public UpdatableLicenseState(Settings settings) {
      super(settings);
    }

    @Override
    public void update(License.OperationMode mode, boolean active) {
      super.update(mode, active);
    }
  }

  public static void putLicense(MetaData.Builder builder, License license) {
    builder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(license, null));
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import static org.elasticsearch.common.unit.TimeValue.timeValueHours;
import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.not;
import static org.mockito.Matchers.any;
import static org.mockito.Mockito.times;
import static org.mockito.Mockito.verify;

public class LicensesAcknowledgementTests extends AbstractLicenseServiceTestCase {

  public void testAcknowledgment() throws Exception {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    setInitialState(TestUtils.generateSignedLicense("gold", timeValueHours(2)), licenseState, Settings.EMPTY);
    licenseService.start();
    // try installing a signed license
    long issueDate = System.currentTimeMillis() - TimeValue.timeValueHours(24 * 2).getMillis();
    License signedLicense = TestUtils.generateSignedLicense("trial", License.VERSION_CURRENT, issueDate,
timeValueHours(10));
    PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(signedLicense);
    // ensure acknowledgement message was part of the response
    licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(false,
LicensesStatus.VALID, true));
    assertThat(licenseService.getLicense(), not(signedLicense));
    verify(clusterService, times(0)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

    // try installing a signed license with acknowledgement
    putLicenseRequest = new PutLicenseRequest().license(signedLicense).acknowledge(true);
    // ensure license was installed and no acknowledgment message was returned
    licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(true,
LicensesStatus.VALID, false));
    verify(clusterService, times(1)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
  }

  public void testRejectUpgradeToProductionWithoutTLS() throws Exception {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    setInitialState(TestUtils.generateSignedLicense("trial", timeValueHours(2)), licenseState, Settings.EMPTY);
    licenseService.start();
    // try installing a signed license
    License signedLicense = TestUtils.generateSignedLicense("platinum", timeValueHours(10));
    PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(signedLicense);
    // ensure acknowledgement message was part of the response
    IllegalStateException ise = expectThrows(IllegalStateException.class, () ->
        licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(false,
LicensesStatus.VALID, true)));
    assertEquals("Cannot install a [PLATINUM] license unless TLS is configured or security is disabled",
ise.getMessage());
  }

  public void testUpgradeToProductionWithoutTLSAndSecurityDisabled() throws Exception {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    setInitialState(TestUtils.generateSignedLicense("trial", timeValueHours(2)), licenseState, Settings.builder()
        .put("xpack.security.enabled", false).build());
    licenseService.start();
    // try installing a signed license
    License signedLicense = TestUtils.generateSignedLicense("platinum", timeValueHours(10));
    PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(signedLicense);
    licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(false,
LicensesStatus.VALID, true));
    assertThat(licenseService.getLicense(), not(signedLicense));
    verify(clusterService, times(1)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

    // try installing a signed license with acknowledgement
    putLicenseRequest = new PutLicenseRequest().license(signedLicense).acknowledge(true);
    // ensure license was installed and no acknowledgment message was returned
    licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(true,
LicensesStatus.VALID, false));
    verify(clusterService, times(2)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
  }

  public void testUpgradeToProductionWithTLSAndSecurity() throws Exception {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    setInitialState(TestUtils.generateSignedLicense("trial", timeValueHours(2)), licenseState, Settings.builder()
        .put("xpack.security.enabled", true)
        .put("xpack.security.transport.ssl.enabled", true).build());
    licenseService.start();
    // try installing a signed license
    License signedLicense = TestUtils.generateSignedLicense("platinum", timeValueHours(10));
    PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(signedLicense);
    licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(false,
LicensesStatus.VALID, true));
    assertThat(licenseService.getLicense(), not(signedLicense));
    verify(clusterService, times(1)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

    // try installing a signed license with acknowledgement
    putLicenseRequest = new PutLicenseRequest().license(signedLicense).acknowledge(true);
    // ensure license was installed and no acknowledgment message was returned
    licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(true,
LicensesStatus.VALID, false));
    verify(clusterService, times(2)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
  }

  private static class AssertingLicensesUpdateResponse implements ActionListener<PutLicenseResponse> {
    private final boolean expectedAcknowledgement;
    private final LicensesStatus expectedStatus;
    private final boolean expectAckMessages;

    AssertingLicensesUpdateResponse(boolean expectedAcknowledgement, LicensesStatus expectedStatus,
                                    boolean expectAckMessages) {
        this.expectedAcknowledgement = expectedAcknowledgement;
        this.expectedStatus = expectedStatus;
        this.expectAckMessages = expectAckMessages;
    }

    @Override
    public void onResponse(PutLicenseResponse licensesUpdateResponse) {
        assertThat(licensesUpdateResponse.isAcknowledged(), equalTo(expectedAcknowledgement));
        assertThat(licensesUpdateResponse.status(), equalTo(expectedStatus));
        assertEquals(licensesUpdateResponse.acknowledgeMessages().isEmpty(), expectAckMessages == false);
    }

    @Override
    public void onFailure(Exception throwable) {
        throw new RuntimeException(throwable);
    }
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.ack.ClusterStateUpdateResponse;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESSingleNodeTestCase;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackSettings;
import org.junit.Before;

import java.util.Collection;
import java.util.Collections;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.atomic.AtomicBoolean;

import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.not;

public class LicensesManagerServiceTests extends ESSingleNodeTestCase {

  @Override
  protected Collection<Class<? extends Plugin>> getPlugins() {
    return Collections.singletonList(LocalStateCompositeXPackPlugin.class);
  }

  @Override
  protected Settings nodeSettings() {
    return Settings.builder()
        .put(super.nodeSettings())
        .put(XPackSettings.SECURITY_ENABLED.getKey(), false)
        .put(XPackSettings.MONITORING_ENABLED.getKey(), false)
        .put(XPackSettings.WATCHER_ENABLED.getKey(), false)
        .put(XPackSettings.GRAPH_ENABLED.getKey(), false)
        .put(XPackSettings.MACHINE_LEARNING_ENABLED.getKey(), false)
        .build();
  }

  @Override
  protected boolean resetNodeAfterTest() {
    return true;
  }

  @Before
  public void waitForTrialLicenseToBeGenerated() throws Exception {
    assertBusy(() ->
assertNotNull(getInstanceFromNode(ClusterService.class).state().metaData().custom(LicensesMetaData.TYPE)));
  }

  public void testStoreAndGetLicenses() throws Exception {
    LicenseService licenseService = getInstanceFromNode(LicenseService.class);
    ClusterService clusterService = getInstanceFromNode(ClusterService.class);
    License goldLicense = TestUtils.generateSignedLicense("gold", TimeValue.timeValueHours(1));
    TestUtils.registerAndAckSignedLicenses(licenseService, goldLicense, LicensesStatus.VALID);
    License silverLicense = TestUtils.generateSignedLicense("silver", TimeValue.timeValueHours(2));
    TestUtils.registerAndAckSignedLicenses(licenseService, silverLicense, LicensesStatus.VALID);
    License platinumLicense = TestUtils.generateSignedLicense("platinum", TimeValue.timeValueHours(1));
    TestUtils.registerAndAckSignedLicenses(licenseService, platinumLicense, LicensesStatus.VALID);
    LicensesMetaData licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
    assertThat(licensesMetaData.getLicense(), equalTo(platinumLicense));
    final License getLicenses = licenseService.getLicense();
    assertThat(getLicenses, equalTo(platinumLicense));
  }

  // TODO: Add test/feature blocking the registration of basic license

  public void testEffectiveLicenses() throws Exception {
    final LicenseService licenseService = getInstanceFromNode(LicenseService.class);
    final ClusterService clusterService = getInstanceFromNode(ClusterService.class);
    License goldLicense = TestUtils.generateSignedLicense("gold", TimeValue.timeValueSeconds(5));
    // put gold license
    TestUtils.registerAndAckSignedLicenses(licenseService, goldLicense, LicensesStatus.VALID);
    LicensesMetaData licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
    assertThat(LicenseService.getLicense(licensesMetaData), equalTo(goldLicense));

    License platinumLicense = TestUtils.generateSignedLicense("platinum", TimeValue.timeValueSeconds(3));
    // put platinum license
    TestUtils.registerAndAckSignedLicenses(licenseService, platinumLicense, LicensesStatus.VALID);
    licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
    assertThat(LicenseService.getLicense(licensesMetaData), equalTo(platinumLicense));
  }

  public void testInvalidLicenseStorage() throws Exception {
    LicenseService licenseService = getInstanceFromNode(LicenseService.class);
    ClusterService clusterService = getInstanceFromNode(ClusterService.class);
    License signedLicense = TestUtils.generateSignedLicense(TimeValue.timeValueMinutes(2));

    // modify content of signed license
    License tamperedLicense = License.builder()
            .fromLicenseSpec(signedLicense, signedLicense.signature())
            .expiryDate(signedLicense.expiryDate() + 10 * 24 * 60 * 60 * 1000L)
            .validate()
            .build();

    TestUtils.registerAndAckSignedLicenses(licenseService, tamperedLicense, LicensesStatus.INVALID);

    // ensure that the invalid license never made it to cluster state
    LicensesMetaData licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
    assertThat(licensesMetaData.getLicense(), not(equalTo(tamperedLicense)));
  }

  public void testRemoveLicenses() throws Exception {
    LicenseService licenseService = getInstanceFromNode(LicenseService.class);
    ClusterService clusterService = getInstanceFromNode(ClusterService.class);

    // generate signed licenses
    License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(1));
    TestUtils.registerAndAckSignedLicenses(licenseService, license, LicensesStatus.VALID);
    LicensesMetaData licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
    assertThat(licensesMetaData.getLicense(), not(LicensesMetaData.LICENSE_TOMBSTONE));

    // remove signed licenses
    removeAndAckSignedLicenses(licenseService);
    licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
    assertThat(licensesMetaData.getLicense(), equalTo(LicensesMetaData.LICENSE_TOMBSTONE));
  }

  private void removeAndAckSignedLicenses(final LicenseService licenseService) {
    final CountDownLatch latch = new CountDownLatch(1);
    final AtomicBoolean success = new AtomicBoolean(false);
    licenseService.removeLicense(new DeleteLicenseRequest(), new
ActionListener<ClusterStateUpdateResponse>() {
        @Override
        public void onResponse(ClusterStateUpdateResponse clusterStateUpdateResponse) {
            if (clusterStateUpdateResponse.isAcknowledged()) {
                success.set(true);
            }
            latch.countDown();
        }

        @Override
        public void onFailure(Exception throwable) {
            latch.countDown();
        }
    });
    try {
        latch.await();
    } catch (InterruptedException e) {
        fail(e.getMessage());
    }
    assertThat("remove license(s) failed", success.get(), equalTo(true));
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.threadpool.TestThreadPool;
import org.elasticsearch.watcher.ResourceWatcherService;
import org.junit.After;
import org.junit.Before;

import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.StandardOpenOption;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.atomic.AtomicReference;
import static org.hamcrest.Matchers.equalTo;

public class OperationModeFileWatcherTests extends ESTestCase {
  private ResourceWatcherService watcherService;
  private TestThreadPool threadPool;
  private Path licenseModePath;
  private OperationModeFileWatcher operationModeFileWatcher;
  private AtomicReference<CountDownLatch> onChangeCounter;

  @Before
  public void setup() throws Exception {
    threadPool = new TestThreadPool("license mode file watcher tests");
    Settings settings = Settings.builder()
        .put("resource.reload.interval.high", "10ms")
        .build();
    watcherService = new ResourceWatcherService(settings,
        threadPool);
    watcherService.start();
    licenseModePath = createTempFile();
    onChangeCounter = new AtomicReference<>(new CountDownLatch(1));
    operationModeFileWatcher = new OperationModeFileWatcher(watcherService, licenseModePath, logger,
        () -> onChangeCounter.get().countDown());
  }

  @After
  public void shutdown() throws InterruptedException {
    terminate(threadPool);
    watcherService.stop();
  }

  public void testInit() throws Exception {
    onChangeCounter.set(new CountDownLatch(2));
    writeMode("gold");
    assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
    operationModeFileWatcher.init();
    assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
    assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.GOLD));
  }

  public void testUpdateModeFromFile() throws Exception {
    Files.delete(licenseModePath);
    operationModeFileWatcher.init();
    assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
    writeMode("gold");
    assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
    assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.GOLD));
```

```
        onChangeCounter.set(new CountDownLatch(1));
        writeMode("basic");
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.BASIC));
    }

    public void testDeleteModeFromFile() throws Exception {
        Files.delete(licenseModePath);
        operationModeFileWatcher.init();
        writeMode("gold");
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.GOLD));
        onChangeCounter.set(new CountDownLatch(1));

        Files.delete(licenseModePath);
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
    }

    public void testInvalidModeFromFile() throws Exception {
        writeMode("invalid");
        operationModeFileWatcher.init();
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
        operationModeFileWatcher.onFileChanged(licenseModePath);
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
    }

    public void testLicenseModeFileIsDirectory() throws Exception {
        licenseModePath = createTempDir();
        operationModeFileWatcher.init();
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
        operationModeFileWatcher.onFileChanged(licenseModePath);
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
    }

    public void testLicenseModeFileCreatedAfterInit() throws Exception {
        Files.delete(licenseModePath);
        operationModeFileWatcher.init();
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
        onChangeCounter.set(new CountDownLatch(1));
        Path tempFile = createTempFile();
        writeMode("gold", tempFile);
        licenseModePath = tempFile;
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.GOLD));
    }

    private void writeMode(String mode) throws IOException {
        writeMode(mode, licenseModePath);
    }

    static void writeMode(String mode, Path file) throws IOException {
        Files.write(file, mode.getBytes(StandardCharsets.UTF_8), StandardOpenOption.CREATE);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.test.ESTestCase;

import java.util.Locale;

import static org.elasticsearch.license.License.OperationMode;
import static org.hamcrest.Matchers.equalTo;

/**
* Tests {@link License.OperationMode} for correctness.
* <p>
* If you change the behavior of these tests, then it means that licensing changes across the products!
*/
public class LicenseOperationModeTests extends ESTestCase {
    public void testResolveTrial() {
        // assert 1.x BWC
        assertResolve(OperationMode.TRIAL, "nONE", "DEv", "deveLopment");
        // assert expected (2.x+) variant
        assertResolve(OperationMode.TRIAL, "tRiAl", "trial");
    }

    public void testResolveBasic() {
        // assert expected (2.x+) variant (note: no 1.x variant of BASIC)
        assertResolve(OperationMode.BASIC, "bAsIc", "basic");
    }

    public void testResolveStandard() {
        // assert expected (2.x+) variant (note: no 1.x variant of STANDARD)
        assertResolve(OperationMode.STANDARD, "StAnDARd", "standard");
    }

    public void testResolveGold() {
        // assert expected (2.x+) variant (note: no different 1.x variant of GOLD)
        assertResolve(OperationMode.GOLD, "SiLvEr", "gOlD", "silver", "gold");
    }

    public void testResolvePlatinum() {
        // assert 1.x BWC
        assertResolve(OperationMode.PLATINUM, "iNtErNaL");
        // assert expected (2.x+) variant
        assertResolve(OperationMode.PLATINUM, "PlAtINum", "platinum");
    }

    public void testResolveUnknown() {
        // 'enterprise' is a type that exists in cloud but should be rejected under normal operation
        // See https://github.com/elastic/x-plugins/issues/3371
        String[] types = { "unknown", "fake", "enterprise" };

        for (String type : types) {
            try {
                OperationMode.resolve(type);

                fail(String.format(Locale.ROOT, "[%s] should not be recognized as an operation mode", type));
            }
            catch (IllegalArgumentException e) {
                assertThat(e.getMessage(), equalTo("unknown type [" + type + "]"));
            }
        }
    }

    private static void assertResolve(OperationMode expected, String... types) {
        for (String type : types) {
            assertThat(OperationMode.resolve(type), equalTo(expected));
        }
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.analysis.common.CommonAnalysisPlugin;
import org.elasticsearch.common.network.NetworkModule;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;

import java.util.Arrays;
import java.util.Collection;

import static org.elasticsearch.test.ESIntegTestCase.Scope.TEST;

@ESIntegTestCase.ClusterScope(scope = TEST, numDataNodes = 0, numClientNodes = 0, maxNumDataNodes =
0, transportClientRatio = 0,
    autoMinMasterNodes = false)
public class LicenseServiceClusterNotRecoveredTests extends AbstractLicensesIntegrationTestCase {

  @Override
  protected Settings nodeSettings(int nodeOrdinal) {
    return nodeSettingsBuilder(nodeOrdinal).build();
  }

  private Settings.Builder nodeSettingsBuilder(int nodeOrdinal) {
    return Settings.builder()
        .put(super.nodeSettings(nodeOrdinal))
        .put("node.data", true)
        .put("resource.reload.interval.high", "500ms") // for license mode file watcher
        .put(NetworkModule.HTTP_ENABLED.getKey(), true);
  }

  @Override
  protected Collection<Class<? extends Plugin>> nodePlugins() {
    return Arrays.asList(LocalStateCompositeXPackPlugin.class, CommonAnalysisPlugin.class,
Netty4Plugin.class);
  }

  @Override
  protected Collection<Class<? extends Plugin>> transportClientPlugins() {
    return nodePlugins();
  }

  public void testClusterNotRecovered() throws Exception {
    logger.info("--> start one master out of two [recovery state]");
    internalCluster().startNode(nodeSettingsBuilder(0).put("discovery.zen.minimum_master_nodes",
2).put("node.master", true));
    logger.info("--> start second master out of two [recovered state]");
    internalCluster().startNode(nodeSettingsBuilder(1).put("discovery.zen.minimum_master_nodes",
2).put("node.master", true));
    assertLicenseActive(true);
  }
}
```

```
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.cluster.ClusterModule;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.metadata.RepositoriesMetaData;
import org.elasticsearch.cluster.metadata.RepositoryMetaData;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.ToXContent.Params;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.test.ESTestCase;

import java.util.Collections;
import java.util.UUID;
import java.util.stream.Collectors;
import java.util.stream.Stream;

import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.notNullValue;
import static org.hamcrest.Matchers.nullValue;

public class LicensesMetaDataSerializationTests extends ESTestCase {

  public void testXContentSerializationOneSignedLicense() throws Exception {
    License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(2));
    LicensesMetaData licensesMetaData = new LicensesMetaData(license, null);
    XContentBuilder builder = XContentFactory.jsonBuilder();
    builder.startObject();
    builder.startObject("licenses");
    licensesMetaData.toXContent(builder, ToXContent.EMPTY_PARAMS);
    builder.endObject();
    builder.endObject();
    LicensesMetaData licensesMetaDataFromXContent =
getLicensesMetaDataFromXContent(createParser(builder));
    assertThat(licensesMetaDataFromXContent.getLicense(), equalTo(license));
    assertNull(licensesMetaDataFromXContent.getMostRecentTrialVersion());
  }
  public void testXContentSerializationOneSignedLicenseWithUsedTrial() throws Exception {
    License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(2));
    LicensesMetaData licensesMetaData = new LicensesMetaData(license, Version.CURRENT);
    XContentBuilder builder = XContentFactory.jsonBuilder();
    builder.startObject();
    builder.startObject("licenses");
    licensesMetaData.toXContent(builder, ToXContent.EMPTY_PARAMS);
    builder.endObject();
    builder.endObject();
    LicensesMetaData licensesMetaDataFromXContent =
getLicensesMetaDataFromXContent(createParser(builder));
    assertThat(licensesMetaDataFromXContent.getLicense(), equalTo(license));
    assertEquals(licensesMetaDataFromXContent.getMostRecentTrialVersion(), Version.CURRENT);
  }

  public void testLicenseMetadataParsingDoesNotSwallowOtherMetaData() throws Exception {
    new Licensing(Settings.EMPTY); // makes sure LicensePlugin is registered in Custom MetaData
    License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(2));
    LicensesMetaData licensesMetaData = new LicensesMetaData(license, Version.CURRENT);
    RepositoryMetaData repositoryMetaData = new RepositoryMetaData("repo", "fs", Settings.EMPTY);
    RepositoriesMetaData repositoriesMetaData = new RepositoriesMetaData(repositoryMetaData);
    final MetaData.Builder metaDataBuilder = MetaData.builder();
    if (randomBoolean()) { // random order of insertion
      metaDataBuilder.putCustom(licensesMetaData.getWriteableName(), licensesMetaData);
      metaDataBuilder.putCustom(repositoriesMetaData.getWriteableName(), repositoriesMetaData);
    } else {
      metaDataBuilder.putCustom(repositoriesMetaData.getWriteableName(), repositoriesMetaData);
      metaDataBuilder.putCustom(licensesMetaData.getWriteableName(), licensesMetaData);
    }
    // serialize metadata
    XContentBuilder builder = XContentFactory.jsonBuilder();
    Params params = new
ToXContent.MapParams(Collections.singletonMap(MetaData.CONTEXT_MODE_PARAM,
MetaData.CONTEXT_MODE_GATEWAY));
    builder.startObject();
    builder = metaDataBuilder.build().toXContent(builder, params);
    builder.endObject();
    // deserialize metadata again
    MetaData metaData = MetaData.Builder.fromXContent(createParser(builder));
    // check that custom metadata still present
    assertThat(metaData.custom(licensesMetaData.getWriteableName()), notNullValue());
    assertThat(metaData.custom(repositoriesMetaData.getWriteableName()), notNullValue());
  }

  public void testXContentSerializationOneTrial() throws Exception {
    long issueDate = System.currentTimeMillis();
    License.Builder specBuilder = License.builder()
        .uid(UUID.randomUUID().toString())
```

```
                .issuedTo("customer")
                .maxNodes(5)
                .issueDate(issueDate)
                .type(randomBoolean() ? "trial" : "basic")
                .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis());
        final License trialLicense = SelfGeneratedLicense.create(specBuilder);
        LicensesMetaData licensesMetaData = new LicensesMetaData(trialLicense, Version.CURRENT);
        XContentBuilder builder = XContentFactory.jsonBuilder();
        builder.startObject();
        builder.startObject("licenses");
        licensesMetaData.toXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.endObject();
        builder.endObject();
        LicensesMetaData licensesMetaDataFromXContent =
getLicensesMetaDataFromXContent(createParser(builder));
        assertThat(licensesMetaDataFromXContent.getLicense(), equalTo(trialLicense));
        assertEquals(licensesMetaDataFromXContent.getMostRecentTrialVersion(), Version.CURRENT);
    }

    public void testLicenseTombstoneFromXContext() throws Exception {
        final XContentBuilder builder = XContentFactory.jsonBuilder();
        builder.startObject();
        builder.startObject("licenses");
        builder.nullField("license");
        builder.endObject();
        builder.endObject();
        LicensesMetaData metaDataFromXContent = getLicensesMetaDataFromXContent(createParser(builder));
        assertThat(metaDataFromXContent.getLicense(), equalTo(LicensesMetaData.LICENSE_TOMBSTONE));
    }

    public void testLicenseTombstoneWithUsedTrialFromXContext() throws Exception {
        final XContentBuilder builder = XContentFactory.jsonBuilder();
        builder.startObject();
        builder.startObject("licenses");
        builder.nullField("license");
        builder.field("trial_license", Version.CURRENT.toString());
        builder.endObject();
        builder.endObject();
        LicensesMetaData metaDataFromXContent = getLicensesMetaDataFromXContent(createParser(builder));
        assertThat(metaDataFromXContent.getLicense(), equalTo(LicensesMetaData.LICENSE_TOMBSTONE));
        assertEquals(metaDataFromXContent.getMostRecentTrialVersion(), Version.CURRENT);
    }

    private static LicensesMetaData getLicensesMetaDataFromXContent(XContentParser parser) throws Exception {
        parser.nextToken(); // consume null
        parser.nextToken(); // consume "licenses"
        LicensesMetaData licensesMetaDataFromXContent = LicensesMetaData.fromXContent(parser);
        parser.nextToken(); // consume endObject
        assertThat(parser.nextToken(), nullValue());
        return licensesMetaDataFromXContent;
    }

    @Override
    protected NamedXContentRegistry xContentRegistry() {
        return new NamedXContentRegistry(Stream.concat(
                new Licensing(Settings.EMPTY).getNamedXContent().stream(),
                ClusterModule.getNamedXWriteables().stream()
        ).collect(Collectors.toList()));
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.cluster.ClusterName;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.settings.Settings;
import org.mockito.ArgumentCaptor;
import org.mockito.Mockito;

import java.util.UUID;

import static org.elasticsearch.license.TestUtils.dateMath;
import static org.mockito.Matchers.any;
import static org.mockito.Mockito.verify;
import static org.mockito.Mockito.when;

public class LicenseRegistrationTests extends AbstractLicenseServiceTestCase {

    public void testSelfGeneratedTrialLicense() throws Exception {
        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        setInitialState(null, licenseState, Settings.EMPTY, "trial");
        when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
        licenseService.start();

        ClusterState state = ClusterState.builder(new ClusterName("a")).build();
        ArgumentCaptor<ClusterStateUpdateTask> stateUpdater =
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
        verify(clusterService, Mockito.times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
        ClusterState stateWithLicense = stateUpdater.getValue().execute(state);
        LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
        assertNotNull(licenseMetaData);
        assertNotNull(licenseMetaData.getLicense());
        assertFalse(licenseMetaData.isEligibleForTrial());
        assertEquals("trial", licenseMetaData.getLicense().type());
        assertEquals(clock.millis() +
LicenseService.NON_BASIC_SELF_GENERATED_LICENSE_DURATION.millis(),
                licenseMetaData.getLicense().expiryDate());
    }

    public void testSelfGeneratedBasicLicense() throws Exception {
        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        setInitialState(null, licenseState, Settings.EMPTY, "basic");
        when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
        licenseService.start();

        ClusterState state = ClusterState.builder(new ClusterName("a")).build();
        ArgumentCaptor<ClusterStateUpdateTask> stateUpdater =
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
        verify(clusterService, Mockito.times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
        ClusterState stateWithLicense = stateUpdater.getValue().execute(state);
        LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
        assertNotNull(licenseMetaData);
        assertNotNull(licenseMetaData.getLicense());
        assertTrue(licenseMetaData.isEligibleForTrial());
        assertEquals("basic", licenseMetaData.getLicense().type());
        assertEquals(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS,
licenseMetaData.getLicense().expiryDate());
    }

    public void testNonSelfGeneratedBasicLicenseIsReplaced() throws Exception {
        long now = System.currentTimeMillis();
        String uid = UUID.randomUUID().toString();
        final License.Builder builder = License.builder()
                .uid(uid)
                .version(License.VERSION_CURRENT)
                .expiryDate(dateMath("now+2h", now))
                .startDate(now)
                .issueDate(now)
                .type("basic")
                .issuedTo("customer")
                .issuer("elasticsearch")
                .maxNodes(5);
        License license = TestUtils.generateSignedLicense(builder);

        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        setInitialState(license, licenseState, Settings.EMPTY);
        when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
        licenseService.start();
        MetaData.Builder mdBuilder = MetaData.builder();
        mdBuilder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(license, null));
        ClusterState state = ClusterState.builder(new ClusterName("a")).metaData(mdBuilder.build()).build();
        ArgumentCaptor<ClusterStateUpdateTask> stateUpdater = 
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
        verify(clusterService, Mockito.times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
        ClusterState stateWithLicense = stateUpdater.getValue().execute(state);
        LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
        assertNotNull(licenseMetaData);
        assertNotNull(licenseMetaData.getLicense());
        assertTrue(licenseMetaData.isEligibleForTrial());
        assertEquals("basic", licenseMetaData.getLicense().type());
        assertEquals(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS, 
licenseMetaData.getLicense().expiryDate());
        assertEquals(uid, licenseMetaData.getLicense().uid());
    }

    public void testExpiredSelfGeneratedBasicLicenseIsExtended() throws Exception {
        long now = System.currentTimeMillis();
        String uid = UUID.randomUUID().toString();
        License.Builder builder = License.builder()
                .uid(uid)
                .issuedTo("name")
                .maxNodes(1000)
                .issueDate(dateMath("now-10h", now))
                .type("basic")
                .expiryDate(dateMath("now-2h", now));
        License license = SelfGeneratedLicense.create(builder);

        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        setInitialState(license, licenseState, Settings.EMPTY);
        when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
        licenseService.start();

        MetaData.Builder mdBuilder = MetaData.builder();
        mdBuilder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(license, null));
        ClusterState state = ClusterState.builder(new ClusterName("a")).metaData(mdBuilder.build()).build();
        ArgumentCaptor<ClusterStateUpdateTask> stateUpdater = 
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
        verify(clusterService, Mockito.times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
        ClusterState stateWithLicense = stateUpdater.getValue().execute(state);
        LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
        assertNotNull(licenseMetaData);
        assertNotNull(licenseMetaData.getLicense());
        assertTrue(licenseMetaData.isEligibleForTrial());
        assertEquals("basic", licenseMetaData.getLicense().type());
        assertEquals(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS, 
licenseMetaData.getLicense().expiryDate());
    assertEquals(uid, licenseMetaData.getLicense().uid());
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.test.ESTestCase;

import java.io.IOException;
import java.nio.ByteBuffer;
import java.util.Base64;
import java.util.Collections;
import java.util.UUID;

import static org.elasticsearch.license.CryptUtils.encrypt;
import static org.hamcrest.Matchers.equalTo;


public class SelfGeneratedLicenseTests extends ESTestCase {

  public void testBasic() throws Exception {
    long issueDate = System.currentTimeMillis();
    License.Builder specBuilder = License.builder()
        .uid(UUID.randomUUID().toString())
        .issuedTo("customer")
        .maxNodes(5)
        .type(randomBoolean() ? "trial" : "basic")
        .issueDate(issueDate)
        .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis());
    License trialLicense = SelfGeneratedLicense.create(specBuilder);
    assertThat(SelfGeneratedLicense.verify(trialLicense), equalTo(true));
  }

  public void testTampered() throws Exception {
    long issueDate = System.currentTimeMillis();
    License.Builder specBuilder = License.builder()
        .uid(UUID.randomUUID().toString())
```

```
                .issuedTo("customer")
                .type(randomBoolean() ? "trial" : "basic")
                .maxNodes(5)
                .issueDate(issueDate)
                .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis());
        License trialLicense = SelfGeneratedLicense.create(specBuilder);
        final String originalSignature = trialLicense.signature();
        License tamperedLicense = License.builder().fromLicenseSpec(trialLicense, originalSignature)
                .expiryDate(System.currentTimeMillis() + TimeValue.timeValueHours(5).getMillis())
                .build();
        assertThat(SelfGeneratedLicense.verify(trialLicense), equalTo(true));
        assertThat(SelfGeneratedLicense.verify(tamperedLicense), equalTo(false));
    }

    public void testFrom1x() throws Exception {
        long issueDate = System.currentTimeMillis();
        License.Builder specBuilder = License.builder()
                .uid(UUID.randomUUID().toString())
                .issuedTo("customer")
                .type("subscription")
                .subscriptionType("trial")
                .issuer("elasticsearch")
                .feature("")
                .version(License.VERSION_START)
                .maxNodes(5)
                .issueDate(issueDate)
                .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis());
        License pre20TrialLicense = specBuilder.build();
        License license =
SelfGeneratedLicense.create(License.builder().fromPre20LicenseSpec(pre20TrialLicense).type("trial"));
        assertThat(SelfGeneratedLicense.verify(license), equalTo(true));
    }

    public void testTrialLicenseVerifyWithOlderVersion() throws Exception {
        long issueDate = System.currentTimeMillis();
        License.Builder specBuilder = License.builder()
                .issuedTo("customer")
                .maxNodes(5)
                .issueDate(issueDate)
                .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis())
                .feature("")
                .subscriptionType("trial")
                .version(1);
        License trialLicenseV1 = createTrialLicense(specBuilder);
        assertThat(SelfGeneratedLicense.verify(trialLicenseV1), equalTo(true));
    }

    private static License createTrialLicense(License.Builder specBuilder) {
        License spec = specBuilder
                .type(randomBoolean() ? "trial" : "basic")
                .issuer("elasticsearch")
                .uid(UUID.randomUUID().toString())
                .build();
        final String signature;
        try {
            XContentBuilder contentBuilder = XContentFactory.contentBuilder(XContentType.JSON);
            spec.toXContent(contentBuilder, new
ToXContent.MapParams(Collections.singletonMap(License.LICENSE_SPEC_VIEW_MODE, "true")));
            byte[] encrypt = encrypt(BytesReference.toBytes(BytesReference.bytes(contentBuilder)));
            byte[] bytes = new byte[4 + 4 + encrypt.length];
            ByteBuffer byteBuffer = ByteBuffer.wrap(bytes);
            byteBuffer.putInt(-spec.version())
                    .putInt(encrypt.length)
                    .put(encrypt);
            signature = Base64.getEncoder().encodeToString(bytes);
        } catch (IOException e) {
            throw new IllegalStateException(e);
        }
        return License.builder().fromLicenseSpec(spec, signature).build();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionFuture;
import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.node.Node;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESSingleNodeTestCase;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackSettings;

import java.nio.charset.StandardCharsets;
import java.util.Collection;
import java.util.Collections;
import java.util.UUID;

import static org.elasticsearch.license.TestUtils.dateMath;
import static org.elasticsearch.license.TestUtils.generateExpiredNonBasicLicense;
import static org.elasticsearch.license.TestUtils.generateSignedLicense;
import static org.hamcrest.CoreMatchers.equalTo;
import static org.hamcrest.CoreMatchers.not;

public class LicensesTransportTests extends ESSingleNodeTestCase {

   @Override
   protected boolean resetNodeAfterTest() {
      return true;
   }

   @Override
   protected Collection<Class<? extends Plugin>> getPlugins() {
      return Collections.singletonList(LocalStateCompositeXPackPlugin.class);
   }

   @Override
   protected Settings nodeSettings() {
      Settings.Builder newSettings = Settings.builder();
      newSettings.put(super.nodeSettings());
      newSettings.put(XPackSettings.SECURITY_ENABLED.getKey(), false);
//        newSettings.put(XPackSettings.MONITORING_ENABLED.getKey(), false);
//        newSettings.put(XPackSettings.WATCHER_ENABLED.getKey(), false);
      newSettings.put(Node.NODE_DATA_SETTING.getKey(), true);
      return newSettings.build();
   }

   public void testEmptyGetLicense() throws Exception {
      // basic license is added async, we should wait for it
      assertBusy(() -> {
         try {
            final ActionFuture<GetLicenseResponse> getLicenseFuture =
                  new GetLicenseRequestBuilder(client().admin().cluster(), GetLicenseAction.INSTANCE).execute();
            final GetLicenseResponse getLicenseResponse;
            getLicenseResponse = getLicenseFuture.get();
            assertNotNull(getLicenseResponse.license());
            assertThat(getLicenseResponse.license().operationMode(), equalTo(License.OperationMode.BASIC));
         } catch (Exception e) {
            throw new RuntimeException("unexpected exception", e);
         }
      });
   }

   public void testPutLicense() throws Exception {
      License signedLicense = generateSignedLicense(TimeValue.timeValueMinutes(2));

      // put license
      PutLicenseRequestBuilder putLicenseRequestBuilder =
        new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE).setLicense(signedLicense)
            .setAcknowledge(true);
    PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
    assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));

    // get and check license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), equalTo(signedLicense));
  }

  public void testPutLicenseFromString() throws Exception {
    License signedLicense = generateSignedLicense(TimeValue.timeValueMinutes(2));
    String licenseString = TestUtils.dumpLicense(signedLicense);

    // put license source
    PutLicenseRequestBuilder putLicenseRequestBuilder =
        new PutLicenseRequestBuilder(client().admin().cluster(), PutLicenseAction.INSTANCE)
            .setLicense(new BytesArray(licenseString.getBytes(StandardCharsets.UTF_8)),
XContentType.JSON)
            .setAcknowledge(true);
    PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
    assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));

    // get and check license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), equalTo(signedLicense));
  }

  public void testPutInvalidLicense() throws Exception {
    License signedLicense = generateSignedLicense(TimeValue.timeValueMinutes(2));

    // modify content of signed license
    License tamperedLicense = License.builder()
        .fromLicenseSpec(signedLicense, signedLicense.signature())
        .expiryDate(signedLicense.expiryDate() + 10 * 24 * 60 * 60 * 1000L)
        .validate()
        .build();

    PutLicenseRequestBuilder builder = new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE);
    builder.setLicense(tamperedLicense);

    // try to put license (should be invalid)
    final PutLicenseResponse putLicenseResponse = builder.get();
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.INVALID));

    // try to get invalid license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), not(tamperedLicense));
  }

  public void testPutBasicLicenseIsInvalid() throws Exception {
    License signedLicense = generateSignedLicense("basic", License.VERSION_CURRENT, -1,
TimeValue.timeValueMinutes(2));

    PutLicenseRequestBuilder builder = new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE);
    builder.setLicense(signedLicense);

    // try to put license (should be invalid)
    IllegalArgumentException iae = expectThrows(IllegalArgumentException.class, builder::get);
    assertEquals(iae.getMessage(), "Registering basic licenses is not allowed.");

    // try to get invalid license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), not(signedLicense));
  }

  public void testPutExpiredLicense() throws Exception {
    License expiredLicense = generateExpiredNonBasicLicense();
    PutLicenseRequestBuilder builder = new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE);
    builder.setLicense(expiredLicense);
    PutLicenseResponse putLicenseResponse = builder.get();
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.EXPIRED));
    // get license should not return the expired license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), not(expiredLicense));
  }

  public void testPutLicensesSimple() throws Exception {
    License goldSignedLicense = generateSignedLicense("gold", TimeValue.timeValueMinutes(5));
    PutLicenseRequestBuilder putLicenseRequestBuilder =
        new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE).setLicense(goldSignedLicense)
            .setAcknowledge(true);
    PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));
```

```
        GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
        assertThat(getLicenseResponse.license(), equalTo(goldSignedLicense));

        License platinumSignedLicense = generateSignedLicense("platinum", TimeValue.timeValueMinutes(2));
        putLicenseRequestBuilder.setLicense(platinumSignedLicense);
        putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));
        getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
        assertThat(getLicenseResponse.license(), equalTo(platinumSignedLicense));
    }

    public void testRemoveLicensesSimple() throws Exception {
        License goldLicense = generateSignedLicense("gold", TimeValue.timeValueMinutes(5));
        PutLicenseRequestBuilder putLicenseRequestBuilder =
                new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE).setLicense(goldLicense)
                        .setAcknowledge(true);
        PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));
        GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
        assertThat(getLicenseResponse.license(), equalTo(goldLicense));
        // delete all licenses
        DeleteLicenseRequestBuilder deleteLicenseRequestBuilder =
                new DeleteLicenseRequestBuilder(client().admin().cluster(), DeleteLicenseAction.INSTANCE);
        DeleteLicenseResponse deleteLicenseResponse = deleteLicenseRequestBuilder.get();
        assertThat(deleteLicenseResponse.isAcknowledged(), equalTo(true));
        // get licenses (expected no licenses)
        getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
        assertNull(getLicenseResponse.license());
    }

    public void testLicenseIsRejectWhenStartDateLaterThanNow() throws Exception {
        long now = System.currentTimeMillis();
        final License.Builder builder = License.builder()
                .uid(UUID.randomUUID().toString())
                .version(License.VERSION_CURRENT)
                .expiryDate(dateMath("now+2h", now))
                .startDate(dateMath("now+1h", now))
                .issueDate(now)
                .type(License.OperationMode.TRIAL.toString())
                .issuedTo("customer")
                .issuer("elasticsearch")
```

```
            .maxNodes(5);
        License license = TestUtils.generateSignedLicense(builder);

        PutLicenseRequestBuilder putLicenseRequestBuilder =
            new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE).setLicense(license)
                .setAcknowledge(true);
        PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.INVALID));
    }

    public void testLicenseIsAcceptedWhenStartDateBeforeThanNow() throws Exception {
        long now = System.currentTimeMillis();
        final License.Builder builder = License.builder()
            .uid(UUID.randomUUID().toString())
            .version(License.VERSION_CURRENT)
            .expiryDate(dateMath("now+2h", now))
            .startDate(now)
            .issueDate(now)
            .type(License.OperationMode.TRIAL.toString())
            .issuedTo("customer")
            .issuer("elasticsearch")
            .maxNodes(5);
        License license = TestUtils.generateSignedLicense(builder);

        PutLicenseRequestBuilder putLicenseRequestBuilder =
            new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE).setLicense(license)
                .setAcknowledge(true);
        PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.action.support.PlainActionFuture;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.transport.TransportAddress;
import org.elasticsearch.common.unit.TimeValue;

import java.net.InetAddress;

import static java.util.Collections.emptyMap;
import static java.util.Collections.emptySet;
import static org.hamcrest.Matchers.containsString;
import static org.mockito.Matchers.any;
import static org.mockito.Mockito.times;
import static org.mockito.Mockito.verify;

public class LicenseTLSTests extends AbstractLicenseServiceTestCase {

  private InetAddress inetAddress;

  public void testApplyLicenseInDevMode() throws Exception {
    License newLicense = TestUtils.generateSignedLicense(randomFrom("gold", "platinum"),
TimeValue.timeValueHours(24L));
    PutLicenseRequest request = new PutLicenseRequest();
    request.acknowledge(true);
    request.license(newLicense);
    Settings settings = Settings.builder().put("xpack.security.enabled", true).build();
    XPackLicenseState licenseState = new XPackLicenseState(settings);
    inetAddress = InetAddress.getLoopbackAddress();

    setInitialState(null, licenseState, settings);
    licenseService.start();
    PlainActionFuture<PutLicenseResponse> responseFuture = new PlainActionFuture<>();
    licenseService.registerLicense(request, responseFuture);
    verify(clusterService).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

    inetAddress = TransportAddress.META_ADDRESS;
    settings = Settings.builder()
        .put("xpack.security.enabled", true)
        .put("discovery.type", "single-node")
        .build();
    licenseService.stop();
    licenseState = new XPackLicenseState(settings);
    setInitialState(null, licenseState, settings);
    licenseService.start();
    licenseService.registerLicense(request, responseFuture);
    verify(clusterService, times(2)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
  }

  public void testApplyLicenseInProdMode() throws Exception {
    final String licenseType = randomFrom("GOLD", "PLATINUM");
    License newLicense = TestUtils.generateSignedLicense(licenseType, TimeValue.timeValueHours(24L));
    PutLicenseRequest request = new PutLicenseRequest();
```

```
        request.acknowledge(true);
        request.license(newLicense);
        Settings settings = Settings.builder().put("xpack.security.enabled", true).build();
        XPackLicenseState licenseState = new XPackLicenseState(settings);
        inetAddress = TransportAddress.META_ADDRESS;

        setInitialState(null, licenseState, settings);
        licenseService.start();
        PlainActionFuture<PutLicenseResponse> responseFuture = new PlainActionFuture<>();
        IllegalStateException e = expectThrows(IllegalStateException.class, () ->
licenseService.registerLicense(request, responseFuture));
        assertThat(e.getMessage(),
                containsString("Cannot install a [" + licenseType + "] license unless TLS is configured or security is
disabled"));

        settings = Settings.builder().put("xpack.security.enabled", false).build();
        licenseService.stop();
        licenseState = new XPackLicenseState(settings);
        setInitialState(null, licenseState, settings);
        licenseService.start();
        licenseService.registerLicense(request, responseFuture);
        verify(clusterService).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

        settings = Settings.builder()
                .put("xpack.security.enabled", true)
                .put("xpack.security.transport.ssl.enabled", true)
                .build();
        licenseService.stop();
        licenseState = new XPackLicenseState(settings);
        setInitialState(null, licenseState, settings);
        licenseService.start();
        licenseService.registerLicense(request, responseFuture);
        verify(clusterService, times(2)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
    }

    @Override
    protected DiscoveryNode getLocalNode() {
        return new DiscoveryNode("localnode", new TransportAddress(inetAddress, randomIntBetween(9300, 9399)),
                emptyMap(), emptySet(), Version.CURRENT);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentHelper;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.test.ESTestCase;

import java.nio.charset.StandardCharsets;
import java.util.Collections;
import java.util.Map;
import java.util.UUID;

import static org.hamcrest.core.IsEqual.equalTo;
import static org.hamcrest.core.IsNull.notNullValue;
import static org.hamcrest.core.IsNull.nullValue;

public class LicenseSerializationTests extends ESTestCase {
  public void testSimpleIssueExpiryDate() throws Exception {
    long now = System.currentTimeMillis();
    String issueDate = TestUtils.dateMathString("now", now);
    String expiryDate = TestUtils.dateMathString("now+10d/d", now);
    String licenseSpecs = TestUtils.generateLicenseSpecString(new TestUtils.LicenseSpec(issueDate, expiryDate));
    License generatedLicense = License.fromSource(new
BytesArray(licenseSpecs.getBytes(StandardCharsets.UTF_8)), XContentType.JSON);
    assertThat(generatedLicense.issueDate(), equalTo(DateUtils.beginningOfTheDay(issueDate)));
    assertThat(generatedLicense.expiryDate(), equalTo(DateUtils.endOfTheDay(expiryDate)));
  }

  public void testLicensesFields() throws Exception {
    TestUtils.LicenseSpec randomLicenseSpec =
TestUtils.generateRandomLicenseSpec(License.VERSION_START);
    String licenseSpecsSource = TestUtils.generateLicenseSpecString(randomLicenseSpec);
    final License fromSource =
        License.fromSource(new BytesArray(licenseSpecsSource.getBytes(StandardCharsets.UTF_8)),
XContentType.JSON);
    TestUtils.assertLicenseSpec(randomLicenseSpec, fromSource);
  }

  public void testLicenseRestView() throws Exception {
    long now = System.currentTimeMillis();
    String expiredLicenseExpiryDate = TestUtils.dateMathString("now-1d/d", now);
    String validLicenseIssueDate = TestUtils.dateMathString("now-10d/d", now);
    String invalidLicenseIssueDate = TestUtils.dateMathString("now+1d/d", now);
    String validLicenseExpiryDate = TestUtils.dateMathString("now+2d/d", now);

    License license = TestUtils.generateLicenses(new TestUtils.LicenseSpec(validLicenseIssueDate,
expiredLicenseExpiryDate));
    XContentBuilder builder = XContentFactory.contentBuilder(XContentType.JSON);
    license.toXContent(builder, new
ToXContent.MapParams(Collections.singletonMap(License.REST_VIEW_MODE, "true")));
    builder.flush();
    Map<String, Object> map = XContentHelper.convertToMap(BytesReference.bytes(builder), false,
builder.contentType()).v2();

    // should have an extra status field, human readable issue_data and expiry_date
    assertThat(map.get("status"), notNullValue());
    assertThat(map.get("issue_date"), notNullValue());
    assertThat(map.get("expiry_date"), notNullValue());
    assertThat(map.get("status"), equalTo("expired"));
    builder = XContentFactory.contentBuilder(XContentType.JSON);
    license.toXContent(builder, ToXContent.EMPTY_PARAMS);
    builder.flush();
    map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();
    assertThat(map.get("status"), nullValue());

    license = TestUtils.generateLicenses(new TestUtils.LicenseSpec(validLicenseIssueDate,
validLicenseExpiryDate));
    builder = XContentFactory.contentBuilder(XContentType.JSON);
    license.toXContent(builder, new
ToXContent.MapParams(Collections.singletonMap(License.REST_VIEW_MODE, "true")));
    builder.flush();
    map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();

    // should have an extra status field, human readable issue_data and expiry_date
    assertThat(map.get("status"), notNullValue());
    assertThat(map.get("issue_date"), notNullValue());
    assertThat(map.get("expiry_date"), notNullValue());
    assertThat(map.get("status"), equalTo("active"));
    builder = XContentFactory.contentBuilder(XContentType.JSON);
    license.toXContent(builder, ToXContent.EMPTY_PARAMS);
    builder.flush();
    map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();
    assertThat(map.get("status"), nullValue());

    license = TestUtils.generateLicenses(new TestUtils.LicenseSpec(invalidLicenseIssueDate,
validLicenseExpiryDate));
    builder = XContentFactory.contentBuilder(XContentType.JSON);
    license.toXContent(builder, new
ToXContent.MapParams(Collections.singletonMap(License.REST_VIEW_MODE, "true")));
    builder.flush();
    map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();

    // should have an extra status field, human readable issue_data and expiry_date
    assertThat(map.get("status"), notNullValue());
    assertThat(map.get("issue_date"), notNullValue());
    assertThat(map.get("expiry_date"), notNullValue());
    assertThat(map.get("status"), equalTo("invalid"));
    builder = XContentFactory.contentBuilder(XContentType.JSON);
    license.toXContent(builder, ToXContent.EMPTY_PARAMS);
    builder.flush();
    map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();
    assertThat(map.get("status"), nullValue());
  }

  public void testLicenseRestViewNonExpiringBasic() throws Exception {
    long now = System.currentTimeMillis();

    License.Builder specBuilder = License.builder()
        .uid(UUID.randomUUID().toString())
        .issuedTo("test")
        .maxNodes(1000)
        .issueDate(now)
        .type("basic")
        .expiryDate(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS);
    License license = SelfGeneratedLicense.create(specBuilder);
    XContentBuilder builder = XContentFactory.contentBuilder(XContentType.JSON);
    license.toXContent(builder, new
ToXContent.MapParams(Collections.singletonMap(License.REST_VIEW_MODE, "true")));
    builder.flush();
    Map<String, Object> map = XContentHelper.convertToMap(BytesReference.bytes(builder), false,
builder.contentType()).v2();

    // should have an extra status field, human readable issue_data and no expiry_date
    assertThat(map.get("status"), notNullValue());
    assertThat(map.get("type"), equalTo("basic"));
    assertThat(map.get("issue_date"), notNullValue());
    assertThat(map.get("expiry_date"), nullValue());
    assertThat(map.get("expiry_date_in_millis"), nullValue());
    assertThat(map.get("status"), equalTo("active"));
    builder = XContentFactory.contentBuilder(XContentType.JSON);
    license.toXContent(builder, ToXContent.EMPTY_PARAMS);
    builder.flush();
    map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();
    assertThat(map.get("status"), nullValue());
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.client.Response;
import org.elasticsearch.client.ResponseException;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.common.io.Streams;
import org.elasticsearch.common.network.NetworkModule;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackClientPlugin;

import java.io.InputStreamReader;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.Collection;

import static org.elasticsearch.test.ESIntegTestCase.Scope.SUITE;

@ESIntegTestCase.ClusterScope(scope = SUITE)
public class StartBasicLicenseTests extends AbstractLicensesIntegrationTestCase {

  @Override
  protected Settings nodeSettings(int nodeOrdinal) {
    return Settings.builder()
        .put(super.nodeSettings(nodeOrdinal))
        .put("node.data", true)
        .put(LicenseService.SELF_GENERATED_LICENSE_TYPE.getKey(), "basic")
        .put(NetworkModule.HTTP_ENABLED.getKey(), true).build();
  }

  @Override
  protected Collection<Class<? extends Plugin>> nodePlugins() {
    return Arrays.asList(LocalStateCompositeXPackPlugin.class, Netty4Plugin.class);
  }

  @Override
  protected Collection<Class<? extends Plugin>> transportClientPlugins() {
    return Arrays.asList(XPackClientPlugin.class, Netty4Plugin.class);
  }

  public void testStartBasicLicense() throws Exception {
    LicensingClient licensingClient = new LicensingClient(client());
    License license = TestUtils.generateSignedLicense("trial",  License.VERSION_CURRENT, -1,
TimeValue.timeValueHours(24));
    licensingClient.preparePutLicense(license).get();
    assertBusy(() -> {
      GetLicenseResponse getLicenseResponse = licensingClient.prepareGetLicense().get();
      assertEquals("trial", getLicenseResponse.license().type());
    });

    // Testing that you can start a basic license when you have no license
    if (randomBoolean()) {
      licensingClient.prepareDeleteLicense().get();
      assertNull(licensingClient.prepareGetLicense().get().license());
    }

    RestClient restClient = getRestClient();
    Response response = restClient.performRequest("GET", "/_xpack/license/basic_status");
    String body = Streams.copyToString(new InputStreamReader(response.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response.getStatusLine().getStatusCode());
    assertEquals("{\"eligible_to_start_basic\":true}", body);

    Response response2 = restClient.performRequest("POST", "/_xpack/license/start_basic?acknowledge=true");
    String body2 = Streams.copyToString(new InputStreamReader(response2.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response2.getStatusLine().getStatusCode());
    assertTrue(body2.contains("\"acknowledged\":true"));
    assertTrue(body2.contains("\"basic_was_started\":true"));

    assertBusy(() -> {
      GetLicenseResponse currentLicense = licensingClient.prepareGetLicense().get();
      assertEquals("basic", currentLicense.license().type());
    });

    long expirationMillis = licensingClient.prepareGetLicense().get().license().expiryDate();
    assertEquals(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS,
expirationMillis);

    Response response3 = restClient.performRequest("GET", "/_xpack/license");
    String body3 = Streams.copyToString(new InputStreamReader(response3.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertTrue(body3.contains("\"type\" : \"basic\""));
    assertFalse(body3.contains("expiry_date"));
    assertFalse(body3.contains("expiry_date_in_millis"));


    Response response4 = restClient.performRequest("GET", "/_xpack/license/basic_status");
    String body4 = Streams.copyToString(new InputStreamReader(response4.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response3.getStatusLine().getStatusCode());
    assertEquals("{\"eligible_to_start_basic\":false}", body4);
    ResponseException ex = expectThrows(ResponseException.class,
        () -> restClient.performRequest("POST", "/_xpack/license/start_basic"));
    Response response5 = ex.getResponse();
    String body5 = Streams.copyToString(new InputStreamReader(response5.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(403, response5.getStatusLine().getStatusCode());
    assertTrue(body5.contains("\"basic_was_started\":false"));
    assertTrue(body5.contains("\"acknowledged\":true"));
    assertTrue(body5.contains("\"error_message\":\"Operation failed: Current license is basic.\""));
  }

  public void testUnacknowledgedStartBasicLicense() throws Exception {
    LicensingClient licensingClient = new LicensingClient(client());
    License license = TestUtils.generateSignedLicense("trial",  License.VERSION_CURRENT, -1,
TimeValue.timeValueHours(24));
    licensingClient.preparePutLicense(license).get();

    assertBusy(() -> {
      GetLicenseResponse getLicenseResponse = licensingClient.prepareGetLicense().get();
      assertEquals("trial", getLicenseResponse.license().type());
    });

    Response response2 = getRestClient().performRequest("POST", "/_xpack/license/start_basic");
    String body2 = Streams.copyToString(new InputStreamReader(response2.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response2.getStatusLine().getStatusCode());
    assertTrue(body2.contains("\"acknowledged\":false"));
    assertTrue(body2.contains("\"basic_was_started\":false"));
    assertTrue(body2.contains("\"error_message\":\"Operation failed: Needs acknowledgement.\""));
    assertTrue(body2.contains("\"message\":\"This license update requires acknowledgement. To acknowledge the
license, " +
        "please read the following messages and call /start_basic again, this time with the
\\\"acknowledge=true\\\""));
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.io.stream.BytesStreamOutput;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentHelper;
import org.elasticsearch.test.ESTestCase;

import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;

import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.not;
import static org.hamcrest.Matchers.sameInstance;

public class PutLicenseResponseTests extends ESTestCase {
  @SuppressWarnings("unchecked")
  public void testSerialization() throws Exception {
    boolean acknowledged = randomBoolean();
    LicensesStatus status = randomFrom(LicensesStatus.VALID, LicensesStatus.INVALID,
LicensesStatus.EXPIRED);
    Map<String, String[]> ackMessages = randomAckMessages();

    PutLicenseResponse response = new PutLicenseResponse(acknowledged, status, "", ackMessages);

    XContentBuilder contentBuilder = XContentFactory.jsonBuilder();
    response.toXContent(contentBuilder, ToXContent.EMPTY_PARAMS);

    Map<String, Object> map = XContentHelper.convertToMap(BytesReference.bytes(contentBuilder), false,
        contentBuilder.contentType()).v2();
    assertThat(map.containsKey("acknowledged"), equalTo(true));
    boolean actualAcknowledged = (boolean) map.get("acknowledged");
    assertThat(actualAcknowledged, equalTo(acknowledged));

    assertThat(map.containsKey("license_status"), equalTo(true));
    String actualStatus = (String) map.get("license_status");
    assertThat(actualStatus, equalTo(status.name().toLowerCase(Locale.ROOT)));

    assertThat(map.containsKey("acknowledge"), equalTo(true));
    Map<String, List<String>> actualAckMessages = (Map<String, List<String>>) map.get("acknowledge");
    assertTrue(actualAckMessages.containsKey("message"));
    actualAckMessages.remove("message");
    assertThat(actualAckMessages.keySet(), equalTo(ackMessages.keySet()));
    for (Map.Entry<String, List<String>> entry : actualAckMessages.entrySet()) {
      assertArrayEquals(entry.getValue().toArray(), ackMessages.get(entry.getKey()));
    }
  }

  public void testStreamSerialization() throws IOException {
    boolean acknowledged = randomBoolean();
    LicensesStatus status = randomFrom(LicensesStatus.VALID, LicensesStatus.INVALID,
LicensesStatus.EXPIRED);
    Map<String, String[]> ackMessages = randomAckMessages();

    // write the steam so that we can attempt to read it back
    BytesStreamOutput output = new BytesStreamOutput();

    PutLicenseResponse response = new PutLicenseResponse(acknowledged, status, "", ackMessages);
    // write it out
    response.writeTo(output);

    StreamInput input = output.bytes().streamInput();

    // read it back in
    response.readFrom(input);

    assertThat(response.isAcknowledged(), equalTo(acknowledged));
    assertThat(response.status(), equalTo(status));
    assertThat(response.acknowledgeMessages(), not(sameInstance(ackMessages)));
    assertThat(response.acknowledgeMessages().size(), equalTo(ackMessages.size()));

    for (String key : ackMessages.keySet()) {
      assertArrayEquals(ackMessages.get(key), response.acknowledgeMessages().get(key));
    }
  }

  private static Map<String, String[]> randomAckMessages() {
    int nFeatures = randomIntBetween(1, 5);

    Map<String, String[]> ackMessages = new HashMap<>();

    for (int i = 0; i < nFeatures; i++) {
      String feature = randomAlphaOfLengthBetween(9, 15);
      int nMessages = randomIntBetween(1, 5);
      String[] messages = new String[nMessages];
      for (int j = 0; j < nMessages; j++) {
        messages[j] = randomAlphaOfLengthBetween(10, 30);
      }
      ackMessages.put(feature, messages);
    }

    return ackMessages;
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.cluster.ClusterName;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlocks;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.component.Lifecycle;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.env.Environment;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.watcher.ResourceWatcherService;
import org.elasticsearch.xpack.core.XPackPlugin;
import org.elasticsearch.xpack.core.watcher.watch.ClockMock;
import org.junit.After;
import org.junit.Before;

import java.nio.file.Path;
import java.util.Arrays;

import static java.util.Collections.emptySet;
import static java.util.Collections.singletonMap;
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.when;

public abstract class AbstractLicenseServiceTestCase extends ESTestCase {

  protected LicenseService licenseService;
  protected ClusterService clusterService;
  protected ResourceWatcherService resourceWatcherService;
  protected ClockMock clock;
  protected DiscoveryNodes discoveryNodes;
  protected Environment environment;
  protected String licenseType;

  @Before
  public void init() throws Exception {
    clusterService = mock(ClusterService.class);
    clock = ClockMock.frozen();
    discoveryNodes = mock(DiscoveryNodes.class);
    resourceWatcherService = mock(ResourceWatcherService.class);
    environment = mock(Environment.class);
  }
  protected void setInitialState(License license, XPackLicenseState licenseState, Settings settings) {
    setInitialState(license, licenseState, settings, randomBoolean() ? "trial" : "basic");
  }

  protected void setInitialState(License license, XPackLicenseState licenseState, Settings settings, String
selfGeneratedType) {
    Path tempDir = createTempDir();
    when(environment.configFile()).thenReturn(tempDir);
    licenseType = selfGeneratedType;
    settings =
Settings.builder().put(settings).put(LicenseService.SELF_GENERATED_LICENSE_TYPE.getKey(),
licenseType).build();
    licenseService = new LicenseService(settings, clusterService, clock, environment, resourceWatcherService,
licenseState);
    ClusterState state = mock(ClusterState.class);
    final ClusterBlocks noBlock = ClusterBlocks.builder().build();
    when(state.blocks()).thenReturn(noBlock);
    MetaData metaData = mock(MetaData.class);
    when(metaData.custom(LicensesMetaData.TYPE)).thenReturn(new LicensesMetaData(license, null));
    when(state.metaData()).thenReturn(metaData);
    final DiscoveryNode mockNode = getLocalNode();
    when(discoveryNodes.getMasterNode()).thenReturn(mockNode);
    when(discoveryNodes.spliterator()).thenReturn(Arrays.asList(mockNode).spliterator());
    when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(false);
    when(state.nodes()).thenReturn(discoveryNodes);
    when(state.getNodes()).thenReturn(discoveryNodes); // it is really ridiculous we have nodes() and getNodes()...
    when(clusterService.state()).thenReturn(state);
    when(clusterService.lifecycleState()).thenReturn(Lifecycle.State.STARTED);
    when(clusterService.getClusterName()).thenReturn(new ClusterName("a"));
    when(clusterService.localNode()).thenReturn(mockNode);
  }

  protected DiscoveryNode getLocalNode() {
    return new DiscoveryNode("b", buildNewFakeTransportAddress(),
singletonMap(XPackPlugin.XPACK_INSTALLED_NODE_ATTR, "true"),
      emptySet(), Version.CURRENT);
  }

  @After
  public void tearDown() throws Exception {
    super.tearDown();
    licenseService.stop();
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import org.elasticsearch.xpack.core.security.authz.privilege.Privilege;

import java.util.function.Predicate;

/**
* A permissions that is based on a general privilege that contains patterns of users that this
* user can execute a request as
*/
public final class RunAsPermission {

  public static final RunAsPermission NONE = new RunAsPermission(Privilege.NONE);

  private final Predicate<String> predicate;

  RunAsPermission(Privilege privilege) {
     this.predicate = privilege.predicate();
  }

  /**
   * Checks if this permission grants run as to the specified user
   */
  public boolean check(String username) {
     return predicate.test(username);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.util.set.Sets;
import org.elasticsearch.xpack.core.security.authz.RoleDescriptor;
import org.elasticsearch.xpack.core.security.authz.accesscontrol.IndicesAccessControl;
import org.elasticsearch.xpack.core.security.authz.privilege.ClusterPrivilege;
import org.elasticsearch.xpack.core.security.authz.privilege.IndexPrivilege;
import org.elasticsearch.xpack.core.security.authz.privilege.Privilege;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.Set;

public final class Role {

  public static final Role EMPTY = Role.builder("__empty").build();

  private final String[] names;
  private final ClusterPermission cluster;
  private final IndicesPermission indices;
  private final RunAsPermission runAs;

  Role(String[] names, ClusterPermission cluster, IndicesPermission indices, RunAsPermission runAs) {
    this.names = names;
    this.cluster = Objects.requireNonNull(cluster);
    this.indices = Objects.requireNonNull(indices);
    this.runAs = Objects.requireNonNull(runAs);
  }

  public String[] names() {
    return names;
  }

  public ClusterPermission cluster() {
    return cluster;
  }

  public IndicesPermission indices() {
    return indices;
  }

  public RunAsPermission runAs() {
    return runAs;
  }

  public static Builder builder(String... names) {
    return new Builder(names, null);
  }

  public static Builder builder(String[] names, FieldPermissionsCache fieldPermissionsCache) {
    return new Builder(names, fieldPermissionsCache);
  }

  public static Builder builder(RoleDescriptor rd, FieldPermissionsCache fieldPermissionsCache) {
    return new Builder(rd, fieldPermissionsCache);
  }
```

```
    /**
     * Returns whether at least one group encapsulated by this indices permissions is authorized to execute the
     * specified action with the requested indices/aliases. At the same time if field and/or document level security
     * is configured for any group also the allowed fields and role queries are resolved.
     */
    public IndicesAccessControl authorize(String action, Set<String> requestedIndicesOrAliases, MetaData
metaData,
                                          FieldPermissionsCache fieldPermissionsCache) {
        Map<String, IndicesAccessControl.IndexAccessControl> indexPermissions = indices.authorize(
                action, requestedIndicesOrAliases, metaData, fieldPermissionsCache
        );

        // At least one role / indices permission set need to match with all the requested indices/aliases:
        boolean granted = true;
        for (Map.Entry<String, IndicesAccessControl.IndexAccessControl> entry : indexPermissions.entrySet()) {
            if (!entry.getValue().isGranted()) {
                granted = false;
                break;
            }
        }
        return new IndicesAccessControl(granted, indexPermissions);
    }

    public static class Builder {

        private final String[] names;
        private ClusterPermission cluster = ClusterPermission.NONE;
        private RunAsPermission runAs = RunAsPermission.NONE;
        private List<IndicesPermission.Group> groups = new ArrayList<>();
        private FieldPermissionsCache fieldPermissionsCache = null;

        private Builder(String[] names, FieldPermissionsCache fieldPermissionsCache) {
            this.names = names;
            this.fieldPermissionsCache = fieldPermissionsCache;
        }

        private Builder(RoleDescriptor rd, @Nullable FieldPermissionsCache fieldPermissionsCache) {
            this.names = new String[] { rd.getName() };
            this.fieldPermissionsCache = fieldPermissionsCache;
            if (rd.getClusterPrivileges().length == 0) {
                cluster = ClusterPermission.NONE;
            } else {
                this.cluster(ClusterPrivilege.get(Sets.newHashSet(rd.getClusterPrivileges())));
            }
            groups.addAll(convertFromIndicesPrivileges(rd.getIndicesPrivileges(), fieldPermissionsCache));
            String[] rdRunAs = rd.getRunAs();
            if (rdRunAs != null && rdRunAs.length > 0) {
```

```
            this.runAs(new Privilege(Sets.newHashSet(rdRunAs), rdRunAs));
        }
    }

    public Builder cluster(ClusterPrivilege privilege) {
        cluster = new ClusterPermission(privilege);
        return this;
    }

    public Builder runAs(Privilege privilege) {
        runAs = new RunAsPermission(privilege);
        return this;
    }

    public Builder add(IndexPrivilege privilege, String... indices) {
        groups.add(new IndicesPermission.Group(privilege, FieldPermissions.DEFAULT, null, indices));
        return this;
    }

    public Builder add(FieldPermissions fieldPermissions, Set<BytesReference> query, IndexPrivilege privilege,
String... indices) {
        groups.add(new IndicesPermission.Group(privilege, fieldPermissions, query, indices));
        return this;
    }

    public Role build() {
        IndicesPermission indices = groups.isEmpty() ? IndicesPermission.NONE :
                new IndicesPermission(groups.toArray(new IndicesPermission.Group[groups.size()]));
        return new Role(names, cluster, indices, runAs);
    }

    static List<IndicesPermission.Group> convertFromIndicesPrivileges(RoleDescriptor.IndicesPrivileges[]
indicesPrivileges,
                                                                    @Nullable FieldPermissionsCache fieldPermissionsCache) {
        List<IndicesPermission.Group> list = new ArrayList<>(indicesPrivileges.length);
        for (RoleDescriptor.IndicesPrivileges privilege : indicesPrivileges) {
            final FieldPermissions fieldPermissions;
            if (fieldPermissionsCache != null) {
                fieldPermissions = fieldPermissionsCache.getFieldPermissions(privilege.getGrantedFields(),
privilege.getDeniedFields());
            } else {
                fieldPermissions = new FieldPermissions(
                        new FieldPermissionsDefinition(privilege.getGrantedFields(), privilege.getDeniedFields()));
            }
            final Set<BytesReference> query = privilege.getQuery() == null ? null :
Collections.singleton(privilege.getQuery());
            list.add(new IndicesPermission.Group(IndexPrivilege.get(Sets.newHashSet(privilege.getPrivileges())),
                    fieldPermissions,
                    query,
                    privilege.getIndices()));

            }
            return list;
        }
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import org.apache.lucene.util.automaton.Automaton;
import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.common.cache.Cache;
import org.elasticsearch.common.cache.CacheBuilder;
import org.elasticsearch.common.settings.Setting;
import org.elasticsearch.common.settings.Setting.Property;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.xpack.core.security.authz.permission.FieldPermissionsDefinition.FieldGrantExcludeGroup;
import org.elasticsearch.xpack.core.security.support.Automatons;

import java.util.Collection;
import java.util.List;
import java.util.Optional;
import java.util.Set;
import java.util.concurrent.ExecutionException;
import java.util.function.Predicate;
import java.util.stream.Collectors;

import static org.elasticsearch.xpack.core.security.SecurityField.setting;

/**
* A service for managing the caching of {@link FieldPermissions} as these may often need to be combined or
created and internally they
* use an {@link org.apache.lucene.util.automaton.Automaton}, which can be costly to create once you account for
minimization
*/
public final class FieldPermissionsCache {

    public static final Setting<Long> CACHE_SIZE_SETTING = Setting.longSetting(
            setting("authz.store.roles.field_permissions.cache.max_size_in_bytes"), 100 * 1024 * 1024, -1L,
Property.NodeScope);
    private final Cache<FieldPermissionsDefinition, FieldPermissions> cache;
    public FieldPermissionsCache(Settings settings) {
        this.cache = CacheBuilder.<FieldPermissionsDefinition, FieldPermissions>builder()
                .setMaximumWeight(CACHE_SIZE_SETTING.get(settings))
                .weigher((key, fieldPermissions) -> fieldPermissions.ramBytesUsed())
                .build();
    }

    /**
     * Gets a {@link FieldPermissions} instance that corresponds to the granted and denied parameters. The instance
may come from the cache
     * or if it gets created, the instance will be cached
     */
    FieldPermissions getFieldPermissions(String[] granted, String[] denied) {
        return getFieldPermissions(new FieldPermissionsDefinition(granted, denied));
    }

    /**
     * Gets a {@link FieldPermissions} instance that corresponds to the granted and denied parameters. The instance
may come from the cache
     * or if it gets created, the instance will be cached
     */
    public FieldPermissions getFieldPermissions(FieldPermissionsDefinition fieldPermissionsDefinition) {
        try {
            return cache.computeIfAbsent(fieldPermissionsDefinition,
                    (key) -> new FieldPermissions(key, FieldPermissions.initializePermittedFieldsAutomaton(key)));
        } catch (ExecutionException e) {
            throw new ElasticsearchException("unable to compute field permissions", e);
        }
    }

    /**
     * Returns a field permissions object that corresponds to the merging of the given field permissions and caches the
instance if one was
     * not found in the cache.
     */
    FieldPermissions getFieldPermissions(Collection<FieldPermissions> fieldPermissionsCollection) {
        Optional<FieldPermissions> allowAllFieldPermissions = fieldPermissionsCollection.stream()
                .filter(((Predicate<FieldPermissions>) (FieldPermissions::hasFieldLevelSecurity)).negate())
                .findFirst();
        return allowAllFieldPermissions.orElseGet(() -> {
            final Set<FieldGrantExcludeGroup> fieldGrantExcludeGroups = fieldPermissionsCollection.stream()
                    .flatMap(fieldPermission ->
fieldPermission.getFieldPermissionsDefinition().getFieldGrantExcludeGroups().stream())
                    .collect(Collectors.toSet());
            final FieldPermissionsDefinition combined = new FieldPermissionsDefinition(fieldGrantExcludeGroups);
            try {
                return cache.computeIfAbsent(combined, (key) -> {
                    List<Automaton> automatonList = fieldPermissionsCollection.stream()
```

```
                                .map(FieldPermissions::getIncludeAutomaton)
                                .collect(Collectors.toList());
                        return new FieldPermissions(key, Automatons.unionAndMinimize(automatonList));
                    });
                } catch (ExecutionException e) {
                    throw new ElasticsearchException("unable to compute field permissions", e);
                }
            });
        }
    }
```

/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/

```
package org.elasticsearch.xpack.core.security.authz.permission;

import org.apache.lucene.util.automaton.Automaton;
import org.apache.lucene.util.automaton.TooComplexToDeterminizeException;
import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.cluster.metadata.AliasOrIndex;
import org.elasticsearch.cluster.metadata.IndexMetaData;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.logging.Loggers;
import org.elasticsearch.xpack.core.security.authz.accesscontrol.IndicesAccessControl;
import org.elasticsearch.xpack.core.security.authz.privilege.IndexPrivilege;
import org.elasticsearch.xpack.core.security.support.Automatons;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.Set;
import java.util.SortedMap;
import java.util.concurrent.ConcurrentHashMap;
import java.util.function.Function;
import java.util.function.Predicate;

import static java.util.Collections.unmodifiableMap;
import static java.util.Collections.unmodifiableSet;
```

```
/**
* A permission that is based on privileges for index related actions executed
* on specific indices
*/
public final class IndicesPermission implements Iterable<IndicesPermission.Group> {

  public static final IndicesPermission NONE = new IndicesPermission();

  private final Function<String, Predicate<String>> loadingFunction;

  private final ConcurrentHashMap<String, Predicate<String>> allowedIndicesMatchersForAction = new
ConcurrentHashMap<>();

  private final Group[] groups;

  public IndicesPermission(Group... groups) {
    this.groups = groups;
    loadingFunction = (action) -> {
      List<String> indices = new ArrayList<>();
      for (Group group : groups) {
        if (group.actionMatcher.test(action)) {
          indices.addAll(Arrays.asList(group.indices));
        }
      }
      return indexMatcher(indices);
    };
  }

  static Predicate<String> indexMatcher(List<String> indices) {
    try {
      return Automatons.predicate(indices);
    } catch (TooComplexToDeterminizeException e) {
      Loggers.getLogger(IndicesPermission.class).debug("Index pattern automaton [{}] is too complex", indices);
      String description = Strings.collectionToCommaDelimitedString(indices);
      if (description.length() > 80) {
        description = Strings.cleanTruncate(description, 80) + "...";
      }
      throw new ElasticsearchSecurityException("The set of permitted index patterns [{}] is too complex to
evaluate", e, description);
    }
  }

  @Override
  public Iterator<Group> iterator() {
    return Arrays.asList(groups).iterator();
  }

  public Group[] groups() {
    return groups;
  }

  /**
   * @return A predicate that will match all the indices that this permission
   * has the privilege for executing the given action on.
   */
  public Predicate<String> allowedIndicesMatcher(String action) {
    return allowedIndicesMatchersForAction.computeIfAbsent(action, loadingFunction);
  }

  /**
   * Checks if the permission matches the provided action, without looking at indices.
   * To be used in very specific cases where indices actions need to be authorized regardless of their indices.
   * The usecase for this is composite actions that are initially only authorized based on the action name (indices are
not
   * checked on the coordinating node), and properly authorized later at the shard level checking their indices as
well.
   */
  public boolean check(String action) {
    for (Group group : groups) {
      if (group.check(action)) {
        return true;
      }
    }
    return false;
  }

  public Automaton allowedActionsMatcher(String index) {
    List<Automaton> automatonList = new ArrayList<>();
    for (Group group : groups) {
      if (group.indexNameMatcher.test(index)) {
        automatonList.add(group.privilege.getAutomaton());
      }
    }
    return automatonList.isEmpty() ? Automatons.EMPTY : Automatons.unionAndMinimize(automatonList);
  }

  /**
   * Authorizes the provided action against the provided indices, given the current cluster metadata
   */
  public Map<String, IndicesAccessControl.IndexAccessControl> authorize(String action, Set<String>
requestedIndicesOrAliases,
                                                                          MetaData metaData, FieldPermissionsCache fieldPermissionsCache) {
    // now... every index that is associated with the request, must be granted
    // by at least one indices permission group

    SortedMap<String, AliasOrIndex> allAliasesAndIndices = metaData.getAliasAndIndexLookup();
    Map<String, Set<FieldPermissions>> fieldPermissionsByIndex = new HashMap<>();
    Map<String, DocumentLevelPermissions> roleQueriesByIndex = new HashMap<>();
    Map<String, Boolean> grantedBuilder = new HashMap<>();

    for (String indexOrAlias : requestedIndicesOrAliases) {
      boolean granted = false;
      Set<String> concreteIndices = new HashSet<>();
      AliasOrIndex aliasOrIndex = allAliasesAndIndices.get(indexOrAlias);
      if (aliasOrIndex != null) {
        for (IndexMetaData indexMetaData : aliasOrIndex.getIndices()) {
          concreteIndices.add(indexMetaData.getIndex().getName());
        }
      }

      for (Group group : groups) {
        if (group.check(action, indexOrAlias)) {
          granted = true;
          for (String index : concreteIndices) {
            Set<FieldPermissions> fieldPermissions = fieldPermissionsByIndex.computeIfAbsent(index, (k) ->
new HashSet<>());
            fieldPermissionsByIndex.put(indexOrAlias, fieldPermissions);
            fieldPermissions.add(group.getFieldPermissions());
            DocumentLevelPermissions permissions =
                roleQueriesByIndex.computeIfAbsent(index, (k) -> new DocumentLevelPermissions());
            roleQueriesByIndex.putIfAbsent(indexOrAlias, permissions);
            if (group.hasQuery()) {
              permissions.addAll(group.getQuery());
            } else {
              // if more than one permission matches for a concrete index here and if
              // a single permission doesn't have a role query then DLS will not be
              // applied even when other permissions do have a role query
              permissions.setAllowAll(true);
            }
          }
        }
      }

      if (concreteIndices.isEmpty()) {
        grantedBuilder.put(indexOrAlias, granted);
      } else {
        grantedBuilder.put(indexOrAlias, granted);
        for (String concreteIndex : concreteIndices) {
          grantedBuilder.put(concreteIndex, granted);
        }
      }
    }

    Map<String, IndicesAccessControl.IndexAccessControl> indexPermissions = new HashMap<>();
```

```
        for (Map.Entry<String, Boolean> entry : grantedBuilder.entrySet()) {
            String index = entry.getKey();
            DocumentLevelPermissions permissions = roleQueriesByIndex.get(index);
            final Set<BytesReference> roleQueries;
            if (permissions != null && permissions.isAllowAll() == false) {
                roleQueries = unmodifiableSet(permissions.queries);
            } else {
                roleQueries = null;
            }

            final FieldPermissions fieldPermissions;
            final Set<FieldPermissions> indexFieldPermissions = fieldPermissionsByIndex.get(index);
            if (indexFieldPermissions != null && indexFieldPermissions.isEmpty() == false) {
                fieldPermissions = indexFieldPermissions.size() == 1 ? indexFieldPermissions.iterator().next() :
                        fieldPermissionsCache.getFieldPermissions(indexFieldPermissions);
            } else {
                fieldPermissions = FieldPermissions.DEFAULT;
            }
            indexPermissions.put(index, new IndicesAccessControl.IndexAccessControl(entry.getValue(),
fieldPermissions, roleQueries));
        }
        return unmodifiableMap(indexPermissions);
    }

    public static class Group {
        private final IndexPrivilege privilege;
        private final Predicate<String> actionMatcher;
        private final String[] indices;
        private final Predicate<String> indexNameMatcher;

        public FieldPermissions getFieldPermissions() {
            return fieldPermissions;
        }

        private final FieldPermissions fieldPermissions;
        private final Set<BytesReference> query;

        public Group(IndexPrivilege privilege, FieldPermissions fieldPermissions, @Nullable Set<BytesReference>
query, String... indices) {
            assert indices.length != 0;
            this.privilege = privilege;
            this.actionMatcher = privilege.predicate();
            this.indices = indices;
            this.indexNameMatcher = indexMatcher(Arrays.asList(indices));
            this.fieldPermissions = Objects.requireNonNull(fieldPermissions);
            this.query = query;
        }
        public IndexPrivilege privilege() {
            return privilege;
        }

        public String[] indices() {
            return indices;
        }

        @Nullable
        public Set<BytesReference> getQuery() {
            return query;
        }

        private boolean check(String action) {
            return actionMatcher.test(action);
        }

        private boolean check(String action, String index) {
            assert index != null;
            return check(action) && indexNameMatcher.test(index);
        }

        boolean hasQuery() {
            return query != null;
        }
    }

    private static class DocumentLevelPermissions {

        private Set<BytesReference> queries = null;
        private boolean allowAll = false;

        private void addAll(Set<BytesReference> query) {
            if (allowAll == false) {
                if (queries == null) {
                    queries = new HashSet<>();
                }
                queries.addAll(query);
            }
        }

        private boolean isAllowAll() {
            return allowAll;
        }

        private void setAllowAll(boolean allowAll) {
            this.allowAll = allowAll;
        }
```

```
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import org.apache.lucene.index.DirectoryReader;
import org.apache.lucene.util.Accountable;
import org.apache.lucene.util.RamUsageEstimator;
import org.apache.lucene.util.automaton.Automata;
import org.apache.lucene.util.automaton.Automaton;
import org.apache.lucene.util.automaton.CharacterRunAutomaton;
import org.apache.lucene.util.automaton.MinimizationOperations;
import org.apache.lucene.util.automaton.Operations;
import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.regex.Regex;
import org.elasticsearch.index.mapper.AllFieldMapper;
import org.elasticsearch.xpack.core.security.authz.accesscontrol.FieldSubsetReader;
import org.elasticsearch.xpack.core.security.authz.permission.FieldPermissionsDefinition.FieldGrantExcludeGroup;
import org.elasticsearch.xpack.core.security.support.Automatons;

import java.io.IOException;
import java.util.Arrays;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.stream.Collectors;

import static org.apache.lucene.util.automaton.Operations.subsetOf;

/**
* Stores patterns to fields which access is granted or denied to and maintains an automaton that can be used to check
if permission is
* allowed for a specific field.
* Field permissions are configured via a list of strings that are patterns a field has to match. Two lists determine
whether or
* not a field is granted access to:
* 1. It has to match the patterns in grantedFieldsArray
* 2. it must not match the patterns in deniedFieldsArray
*/
public final class FieldPermissions implements Accountable {

  public static final FieldPermissions DEFAULT = new FieldPermissions();
  private static final long BASE_FIELD_PERM_DEF_BYTES = RamUsageEstimator.shallowSizeOf(new
FieldPermissionsDefinition(null, null));
  private static final long BASE_FIELD_GROUP_BYTES = RamUsageEstimator.shallowSizeOf(new
FieldGrantExcludeGroup(null, null));
  private static final long BASE_HASHSET_SIZE = RamUsageEstimator.shallowSizeOfInstance(HashSet.class);
  private static final long BASE_HASHSET_ENTRY_SIZE;
  static {
    HashMap<String, Object> map = new HashMap<>();
    map.put(FieldPermissions.class.getName(), new Object());
    long mapEntryShallowSize = RamUsageEstimator.shallowSizeOf(map.entrySet().iterator().next());
    // assume a load factor of 50%
    // for each entry, we need two object refs, one for the entry itself
    // and one for the free space that is due to the fact hash tables can
    // not be fully loaded
    BASE_HASHSET_ENTRY_SIZE = mapEntryShallowSize + 2 *
RamUsageEstimator.NUM_BYTES_OBJECT_REF;
  }

  private final FieldPermissionsDefinition fieldPermissionsDefinition;
  // an automaton that represents a union of one more sets of permitted and denied fields
  private final CharacterRunAutomaton permittedFieldsAutomaton;
  private final boolean permittedFieldsAutomatonIsTotal;
  private final Automaton originalAutomaton;

  private final long ramBytesUsed;

  /** Constructor that does not enable field-level security: all fields are accepted. */
  public FieldPermissions() {
    this(new FieldPermissionsDefinition(null, null), Automatons.MATCH_ALL);
  }

  /** Constructor that enables field-level security based on include/exclude rules. Exclude rules
   *  have precedence over include rules. */
  public FieldPermissions(FieldPermissionsDefinition fieldPermissionsDefinition) {
    this(fieldPermissionsDefinition, initializePermittedFieldsAutomaton(fieldPermissionsDefinition));
  }

  /** Constructor that enables field-level security based on include/exclude rules. Exclude rules
   *  have precedence over include rules. */
  FieldPermissions(FieldPermissionsDefinition fieldPermissionsDefinition, Automaton permittedFieldsAutomaton)
{
    if (permittedFieldsAutomaton.isDeterministic() == false && permittedFieldsAutomaton.getNumStates() > 1) {
      // we only accept deterministic automata so that the CharacterRunAutomaton constructor
      // directly wraps the provided automaton
      throw new IllegalArgumentException("Only accepts deterministic automata");
    }
    this.fieldPermissionsDefinition = fieldPermissionsDefinition;
    this.originalAutomaton = permittedFieldsAutomaton;
    this.permittedFieldsAutomaton = new CharacterRunAutomaton(permittedFieldsAutomaton);
    // we cache the result of isTotal since this might be a costly operation
    this.permittedFieldsAutomatonIsTotal = Operations.isTotal(permittedFieldsAutomaton);

    long ramBytesUsed = BASE_FIELD_PERM_DEF_BYTES;

    for (FieldGrantExcludeGroup group : fieldPermissionsDefinition.getFieldGrantExcludeGroups()) {
      ramBytesUsed += BASE_FIELD_GROUP_BYTES + BASE_HASHSET_ENTRY_SIZE;
      if (group.getGrantedFields() != null) {
        ramBytesUsed += RamUsageEstimator.shallowSizeOf(group.getGrantedFields());
      }
      if (group.getExcludedFields() != null) {
        ramBytesUsed += RamUsageEstimator.shallowSizeOf(group.getExcludedFields());
      }
    }
    ramBytesUsed += permittedFieldsAutomaton.ramBytesUsed();
    ramBytesUsed += runAutomatonRamBytesUsed(permittedFieldsAutomaton);
    this.ramBytesUsed = ramBytesUsed;
  }

  /**
   * Return an estimation of the ram bytes used by a {@link CharacterRunAutomaton}
   * that wraps the given automaton.
   */
  private static long runAutomatonRamBytesUsed(Automaton a) {
    return a.getNumStates() * 5; // wild guess, better than 0
  }

  public static Automaton initializePermittedFieldsAutomaton(FieldPermissionsDefinition
fieldPermissionsDefinition) {
    Set<FieldGrantExcludeGroup> groups = fieldPermissionsDefinition.getFieldGrantExcludeGroups();
    assert groups.size() > 0 : "there must always be a single group for field inclusion/exclusion";
    List<Automaton> automatonList =
        groups.stream()
            .map(g -> FieldPermissions.initializePermittedFieldsAutomaton(g.getGrantedFields(),
g.getExcludedFields()))
            .collect(Collectors.toList());
    return Automatons.unionAndMinimize(automatonList);
  }

  private static Automaton initializePermittedFieldsAutomaton(final String[] grantedFields, final String[]
deniedFields) {
    Automaton grantedFieldsAutomaton;
    if (grantedFields == null || Arrays.stream(grantedFields).anyMatch(Regex::isMatchAllPattern)) {
      grantedFieldsAutomaton = Automatons.MATCH_ALL;
    } else {
      // an automaton that includes metadata fields, including join fields created by the _parent field such
```

```
            // as _parent#type
            Automaton metaFieldsAutomaton = Operations.concatenate(Automata.makeChar('_'),
Automata.makeAnyString());
            grantedFieldsAutomaton = Operations.union(Automatons.patterns(grantedFields), metaFieldsAutomaton);
        }

        Automaton deniedFieldsAutomaton;
        if (deniedFields == null || deniedFields.length == 0) {
            deniedFieldsAutomaton = Automatons.EMPTY;
        } else {
            deniedFieldsAutomaton = Automatons.patterns(deniedFields);
        }

        grantedFieldsAutomaton = MinimizationOperations.minimize(grantedFieldsAutomaton,
Operations.DEFAULT_MAX_DETERMINIZED_STATES);
        deniedFieldsAutomaton = MinimizationOperations.minimize(deniedFieldsAutomaton,
Operations.DEFAULT_MAX_DETERMINIZED_STATES);

        if (subsetOf(deniedFieldsAutomaton, grantedFieldsAutomaton) == false) {
            throw new ElasticsearchSecurityException("Exceptions for field permissions must be a subset of the " +
                    "granted fields but " + Strings.arrayToCommaDelimitedString(deniedFields) + " is not a subset of " +
                    Strings.arrayToCommaDelimitedString(grantedFields));
        }

        if (!containsAllField(grantedFields) && !containsAllField(deniedFields)) {
            // It is not explicitly stated whether _all should be allowed/denied
            // In that case we automatically disable _all, unless all fields would match
            if (Operations.isTotal(grantedFieldsAutomaton) && Operations.isEmpty(deniedFieldsAutomaton)) {
                // all fields are accepted, so using _all is fine
            } else {
                deniedFieldsAutomaton = Operations.union(deniedFieldsAutomaton,
Automata.makeString(AllFieldMapper.NAME));
            }
        }

        grantedFieldsAutomaton = Automatons.minusAndMinimize(grantedFieldsAutomaton,
deniedFieldsAutomaton);
        return grantedFieldsAutomaton;
    }

    private static boolean containsAllField(String[] fields) {
        return fields != null && Arrays.stream(fields).anyMatch(AllFieldMapper.NAME::equals);
    }

    /**
     * Returns true if this field permission policy allows access to the field and false if not.
     * fieldName can be a wildcard.
     */
```

```
  public boolean grantsAccessTo(String fieldName) {
    return permittedFieldsAutomatonIsTotal || permittedFieldsAutomaton.run(fieldName);
  }

  FieldPermissionsDefinition getFieldPermissionsDefinition() {
    return fieldPermissionsDefinition;
  }

  /** Return whether field-level security is enabled, ie. whether any field might be filtered out. */
  public boolean hasFieldLevelSecurity() {
    return permittedFieldsAutomatonIsTotal == false;
  }

  /** Return a wrapped reader that only exposes allowed fields. */
  public DirectoryReader filter(DirectoryReader reader) throws IOException {
    if (hasFieldLevelSecurity() == false) {
      return reader;
    }
    return FieldSubsetReader.wrap(reader, permittedFieldsAutomaton);
  }

  // for testing only
  Automaton getIncludeAutomaton() {
    return originalAutomaton;
  }

  @Override
  public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;

    FieldPermissions that = (FieldPermissions) o;

    if (permittedFieldsAutomatonIsTotal != that.permittedFieldsAutomatonIsTotal) return false;
    return fieldPermissionsDefinition != null ?
        fieldPermissionsDefinition.equals(that.fieldPermissionsDefinition) : that.fieldPermissionsDefinition ==
null;
  }

  @Override
  public int hashCode() {
    int result = fieldPermissionsDefinition != null ? fieldPermissionsDefinition.hashCode() : 0;
    result = 31 * result + (permittedFieldsAutomatonIsTotal ? 1 : 0);
    return result;
  }

  @Override
  public long ramBytesUsed() {
        return ramBytesUsed;
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import org.elasticsearch.xpack.core.security.authz.privilege.ClusterPrivilege;

import java.util.function.Predicate;

/**
* A permission that is based on privileges for cluster wide actions
*/
public final class ClusterPermission {

    public static final ClusterPermission NONE = new ClusterPermission(ClusterPrivilege.NONE);

    private final ClusterPrivilege privilege;
    private final Predicate<String> predicate;

    ClusterPermission(ClusterPrivilege privilege) {
        this.privilege = privilege;
        this.predicate = privilege.predicate();
    }

    public ClusterPrivilege privilege() {
        return privilege;
    }

    public boolean check(String action) {
        return predicate.test(action);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import java.util.Arrays;
import java.util.Collections;
import java.util.Set;
```

```
/**
* Represents the definition of a {@link FieldPermissions}. Field permissions are defined as a
* collections of grant and exclude definitions where the exclude definition must be a subset of
* the grant definition.
*/
public final class FieldPermissionsDefinition {

  private final Set<FieldGrantExcludeGroup> fieldGrantExcludeGroups;

  public FieldPermissionsDefinition(String[] grant, String[] exclude) {
    this(Collections.singleton(new FieldGrantExcludeGroup(grant, exclude)));
  }

  public FieldPermissionsDefinition(Set<FieldGrantExcludeGroup> fieldGrantExcludeGroups) {
    this.fieldGrantExcludeGroups = Collections.unmodifiableSet(fieldGrantExcludeGroups);
  }

  public Set<FieldGrantExcludeGroup> getFieldGrantExcludeGroups() {
    return fieldGrantExcludeGroups;
  }

  @Override
  public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;

    FieldPermissionsDefinition that = (FieldPermissionsDefinition) o;

    return fieldGrantExcludeGroups != null ?
        fieldGrantExcludeGroups.equals(that.fieldGrantExcludeGroups) :
        that.fieldGrantExcludeGroups == null;
  }

  @Override
  public int hashCode() {
    return fieldGrantExcludeGroups != null ? fieldGrantExcludeGroups.hashCode() : 0;
  }

  public static final class FieldGrantExcludeGroup {
    private final String[] grantedFields;
    private final String[] excludedFields;

    public FieldGrantExcludeGroup(String[] grantedFields, String[] excludedFields) {
      this.grantedFields = grantedFields;
      this.excludedFields = excludedFields;
    }

    public String[] getGrantedFields() {
            return grantedFields;
        }

        public String[] getExcludedFields() {
            return excludedFields;
        }

        @Override
        public boolean equals(Object o) {
            if (this == o) return true;
            if (o == null || getClass() != o.getClass()) return false;

            FieldGrantExcludeGroup that = (FieldGrantExcludeGroup) o;

            if (!Arrays.equals(grantedFields, that.grantedFields)) return false;
            return Arrays.equals(excludedFields, that.excludedFields);
        }

        @Override
        public int hashCode() {
            int result = Arrays.hashCode(grantedFields);
            result = 31 * result + Arrays.hashCode(excludedFields);
            return result;
        }
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;
import org.elasticsearch.client.ElasticsearchClient;

public class GetLicenseAction extends Action<GetLicenseRequest, GetLicenseResponse,
GetLicenseRequestBuilder> {

    public static final GetLicenseAction INSTANCE = new GetLicenseAction();
    public static final String NAME = "cluster:monitor/xpack/license/get";

    private GetLicenseAction() {
        super(NAME);
    }

    @Override
    public GetLicenseResponse newResponse() {
        return new GetLicenseResponse();
    }

    @Override
    public GetLicenseRequestBuilder newRequestBuilder(ElasticsearchClient client) {
        return new GetLicenseRequestBuilder(client, this);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.cluster.AbstractNamedDiffable;
import org.elasticsearch.cluster.MergableCustomMetaData;
import org.elasticsearch.cluster.NamedDiff;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.io.IOException;
import java.util.EnumSet;

/**
* Contains metadata about registered licenses
*/
public class LicensesMetaData extends AbstractNamedDiffable<MetaData.Custom> implements
XPackPlugin.XPackMetaDataCustom,
        MergableCustomMetaData<LicensesMetaData> {

    public static final String TYPE = "licenses";

    /**
     * When license is explicitly removed by a user, LICENSE_TOMBSTONE
     * is used as a placeholder in the license metadata. This enables
     * us to distinguish between the scenario when a cluster never
     * had a license (null) and when a license was removed explicitly
     * (LICENSE_TOMBSTONE).
     * We rely on this to decide whether to generate a unsigned trial
     * license or not. we should only generate a license if no license
```

```
 * ever existed in the cluster state
 */
public static final License LICENSE_TOMBSTONE = License.builder()
        .type("trial")
        .issuer("elasticsearch")
        .uid("TOMBSTONE")
        .issuedTo("")
        .maxNodes(0)
        .issueDate(0)
        .expiryDate(0)
        .build();

private License license;

// This field describes the version of x-pack for which this cluster has exercised a trial. If the field
// is null, then no trial has been exercised. We keep the version to leave open the possibility that we
// may eventually allow a cluster to exercise a trial every time they upgrade to a new major version.
@Nullable
private Version trialVersion;

LicensesMetaData(License license, Version trialVersion) {
    this.license = license;
    this.trialVersion = trialVersion;
}

public License getLicense() {
    return license;
}

boolean isEligibleForTrial() {
    if (trialVersion == null) {
        return true;
    }
    return Version.CURRENT.major > trialVersion.major;
}

Version getMostRecentTrialVersion() {
    return trialVersion;
}

@Override
public String toString() {
    return "LicensesMetaData{" +
            "license=" + license +
            ", trialVersion=" + trialVersion +
            '}';
}
```

```
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;

    LicensesMetaData that = (LicensesMetaData) o;

    if (license != null ? !license.equals(that.license) : that.license != null) return false;
    return trialVersion != null ? trialVersion.equals(that.trialVersion) : that.trialVersion == null;
}

@Override
public int hashCode() {
    int result = license != null ? license.hashCode() : 0;
    result = 31 * result + (trialVersion != null ? trialVersion.hashCode() : 0);
    return result;
}

@Override
public String getWriteableName() {
    return TYPE;
}

@Override
public Version getMinimalSupportedVersion() {
    return Version.CURRENT.minimumCompatibilityVersion();
}

@Override
public EnumSet<MetaData.XContentContext> context() {
    return EnumSet.of(MetaData.XContentContext.GATEWAY);
}

public static LicensesMetaData fromXContent(XContentParser parser) throws IOException {
    License license = LICENSE_TOMBSTONE;
    Version trialLicense = null;
    XContentParser.Token token;
    while ((token = parser.nextToken()) != XContentParser.Token.END_OBJECT) {
        if (token == XContentParser.Token.FIELD_NAME) {
            String fieldName = parser.currentName();
            if (fieldName != null) {
                if (fieldName.equals(Fields.LICENSE)) {
                    token = parser.nextToken();
                    if (token == XContentParser.Token.START_OBJECT) {
                        license = License.fromXContent(parser);
                    } else if (token == XContentParser.Token.VALUE_NULL) {
                        license = LICENSE_TOMBSTONE;
                    }
                } else if (fieldName.equals(Fields.TRIAL_LICENSE)) {
                    parser.nextToken();
                    trialLicense = Version.fromString(parser.text());
                }
            }
        }
    }
    return new LicensesMetaData(license, trialLicense);
}

@Override
public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
    if (license == LICENSE_TOMBSTONE) {
        builder.nullField(Fields.LICENSE);
    } else {
        builder.startObject(Fields.LICENSE);
        license.toInnerXContent(builder, params);
        builder.endObject();
    }
    if (trialVersion != null) {
        builder.field(Fields.TRIAL_LICENSE, trialVersion.toString());
    }
    return builder;
}

@Override
public void writeTo(StreamOutput streamOutput) throws IOException {
    if (license == LICENSE_TOMBSTONE) {
        streamOutput.writeBoolean(false); // no license
    } else {
        streamOutput.writeBoolean(true); // has a license
        license.writeTo(streamOutput);
    }
    if (streamOutput.getVersion().onOrAfter(Version.V_6_1_0)) {
        if (trialVersion == null) {
            streamOutput.writeBoolean(false);
        } else {
            streamOutput.writeBoolean(true);
            Version.writeVersion(trialVersion, streamOutput);
        }
    }
}

public LicensesMetaData(StreamInput streamInput) throws IOException {
    if (streamInput.readBoolean()) {
        license = License.readLicense(streamInput);
    } else {
        license = LICENSE_TOMBSTONE;
        }
        if (streamInput.getVersion().onOrAfter(Version.V_6_1_0)) {
            boolean hasExercisedTrial = streamInput.readBoolean();
            if (hasExercisedTrial) {
                this.trialVersion = Version.readVersion(streamInput);
            }
        }
    }

    public static NamedDiff<MetaData.Custom> readDiffFrom(StreamInput streamInput) throws IOException {
        return readDiffFrom(MetaData.Custom.class, TYPE, streamInput);
    }

    public static License extractLicense(LicensesMetaData licensesMetaData) {
        if (licensesMetaData != null) {
            License license = licensesMetaData.getLicense();
            if (license == LicensesMetaData.LICENSE_TOMBSTONE) {
                return null;
            } else {
                return license;
            }
        }
        return null;
    }

    @Override
    public LicensesMetaData merge(LicensesMetaData other) {
        if (other.license == null) {
            return this;
        } else if (license == null
                || OperationMode.compare(other.license.operationMode(), license.operationMode()) > 0) {
            return other;
        }
        return this;
    }

    private static final class Fields {
        private static final String LICENSE = "license";
        private static final String TRIAL_LICENSE = "trial_license";
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.action.RestToXContentListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.DELETE;

public class RestDeleteLicenseAction extends XPackRestHandler {
  public RestDeleteLicenseAction(Settings settings, RestController controller) {
     super(settings);
     // @deprecated Remove deprecations in 6.0
     controller.registerWithDeprecatedHandler(DELETE, URI_BASE + "/license", this,
                                        DELETE, "/_license", deprecationLogger);

     // Remove _licenses support entirely in 6.0
     controller.registerAsDeprecatedHandler(DELETE, "/_licenses", this,
                                      "[DELETE /_licenses] is deprecated! Use " +
                                      "[DELETE /_xpack/license] instead.",
                                      deprecationLogger);
  }

  @Override
  public String getName() {
     return "xpack_delete_license_action";
  }

  @Override
  public RestChannelConsumer doPrepareRequest(final RestRequest request, final XPackClient client) throws
IOException {
     DeleteLicenseRequest deleteLicenseRequest = new DeleteLicenseRequest();
     deleteLicenseRequest.timeout(request.paramAsTime("timeout", deleteLicenseRequest.timeout()));
     deleteLicenseRequest.masterNodeTimeout(request.paramAsTime("master_timeout",
deleteLicenseRequest.masterNodeTimeout()));

     return channel -> client.es().admin().cluster().execute(DeleteLicenseAction.INSTANCE, deleteLicenseRequest,
            new RestToXContentListener<>(channel));
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.action.Action;
import org.elasticsearch.client.ElasticsearchClient;

public class PutLicenseAction extends Action<PutLicenseRequest, PutLicenseResponse,
PutLicenseRequestBuilder> {

   public static final PutLicenseAction INSTANCE = new PutLicenseAction();
   public static final String NAME = "cluster:admin/xpack/license/put";

   private PutLicenseAction() {
      super(NAME);
   }

   @Override
   public PutLicenseResponse newResponse() {
      return new PutLicenseResponse();
   }

   @Override
   public PutLicenseRequestBuilder newRequestBuilder(ElasticsearchClient client) {
      return new PutLicenseRequestBuilder(client, this);
   }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.MasterNodeReadRequest;
import org.elasticsearch.common.io.stream.StreamInput;

import java.io.IOException;

public class GetBasicStatusRequest extends MasterNodeReadRequest<GetBasicStatusRequest> {

   public GetBasicStatusRequest() {
   }

   @Override
   public ActionRequestValidationException validate() {
      return null;
   }
}
```

```
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;
import org.elasticsearch.client.ElasticsearchClient;

public class GetTrialStatusAction extends Action<GetTrialStatusRequest, GetTrialStatusResponse,
GetTrialStatusRequestBuilder> {

  public static final GetTrialStatusAction INSTANCE = new GetTrialStatusAction();
  public static final String NAME = "cluster:admin/xpack/license/trial_status";

  private GetTrialStatusAction() {
    super(NAME);
  }

  @Override
  public GetTrialStatusRequestBuilder newRequestBuilder(ElasticsearchClient client) {
    return new GetTrialStatusRequestBuilder(client, this);
  }

  @Override
  public GetTrialStatusResponse newResponse() {
    return new GetTrialStatusResponse();
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeReadAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportGetBasicStatusAction extends TransportMasterNodeReadAction<GetBasicStatusRequest,
GetBasicStatusResponse> {

  @Inject
  public TransportGetBasicStatusAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                       ThreadPool threadPool, ActionFilters actionFilters,
                                       IndexNameExpressionResolver indexNameExpressionResolver) {
    super(settings, GetBasicStatusAction.NAME, transportService, clusterService, threadPool, actionFilters,
        indexNameExpressionResolver, GetBasicStatusRequest::new);
  }

  @Override
  protected String executor() {
    return ThreadPool.Names.SAME;
  }

  @Override
  protected GetBasicStatusResponse newResponse() {
    return new GetBasicStatusResponse();
  }

  @Override
  protected void masterOperation(GetBasicStatusRequest request, ClusterState state,
                                 ActionListener<GetBasicStatusResponse> listener) throws Exception {
    LicensesMetaData licensesMetaData = state.metaData().custom(LicensesMetaData.TYPE);
    if (licensesMetaData == null) {
      listener.onResponse(new GetBasicStatusResponse(true));
    } else {
      License license = licensesMetaData.getLicense();
      listener.onResponse(new GetBasicStatusResponse(license == null || license.type().equals("basic") == false));
    }

  }

  @Override
  protected ClusterBlockException checkBlock(GetBasicStatusRequest request, ClusterState state) {
    return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_READ);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportPostStartBasicAction extends TransportMasterNodeAction<PostStartBasicRequest,
PostStartBasicResponse> {

  private final LicenseService licenseService;

  @Inject
  public TransportPostStartBasicAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                       LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                                       IndexNameExpressionResolver indexNameExpressionResolver) {
    super(settings, PostStartBasicAction.NAME, transportService, clusterService, threadPool, actionFilters,
        indexNameExpressionResolver, PostStartBasicRequest::new);
    this.licenseService = licenseService;
  }

  @Override
  protected String executor() {
    return ThreadPool.Names.SAME;
  }

  @Override
  protected PostStartBasicResponse newResponse() {
    return new PostStartBasicResponse();
  }

  @Override
  protected void masterOperation(PostStartBasicRequest request, ClusterState state,
                                 ActionListener<PostStartBasicResponse> listener) throws Exception {
    licenseService.startBasicLicense(request, listener);
  }

  @Override
  protected ClusterBlockException checkBlock(PostStartBasicRequest request, ClusterState state) {
    return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_WRITE);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.RestStatus;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.GET;

public class RestGetTrialStatus extends XPackRestHandler {

  RestGetTrialStatus(Settings settings, RestController controller) {
    super(settings);
    controller.registerHandler(GET, URI_BASE + "/license/trial_status", this);
  }

  @Override
  protected RestChannelConsumer doPrepareRequest(RestRequest request, XPackClient client) throws IOException
{
    return channel -> client.licensing().prepareGetStartTrial().execute(
        new RestBuilderListener<GetTrialStatusResponse>(channel) {
          @Override
          public RestResponse buildResponse(GetTrialStatusResponse response, XContentBuilder builder) throws
Exception {
            builder.startObject();
            builder.field("eligible_to_start_trial", response.isEligibleToStartTrial());
            builder.endObject();
            return new BytesRestResponse(RestStatus.OK, builder);
          }
        });
  }

  @Override
  public String getName() {
    return "xpack_trial_status_action";
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;
import org.elasticsearch.client.ElasticsearchClient;

public class DeleteLicenseAction extends Action<DeleteLicenseRequest, DeleteLicenseResponse,
DeleteLicenseRequestBuilder> {

  public static final DeleteLicenseAction INSTANCE = new DeleteLicenseAction();
  public static final String NAME = "cluster:admin/xpack/license/delete";

  private DeleteLicenseAction() {
    super(NAME);
  }

  @Override
  public DeleteLicenseResponse newResponse() {
    return new DeleteLicenseResponse();
  }

  @Override
  public DeleteLicenseRequestBuilder newRequestBuilder(ElasticsearchClient client) {
    return new DeleteLicenseRequestBuilder(client, this);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.MasterNodeReadRequest;
import org.elasticsearch.common.io.stream.StreamInput;

import java.io.IOException;

public class GetTrialStatusRequest extends MasterNodeReadRequest<GetTrialStatusRequest> {
  public GetTrialStatusRequest() {
  }

  @Override
  public ActionRequestValidationException validate() {
    return null;
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.apache.lucene.util.BytesRef;
import org.apache.lucene.util.BytesRefIterator;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.core.internal.io.Streams;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.Signature;
import java.security.SignatureException;
import java.util.Arrays;
import java.util.Base64;
import java.util.Collections;

/**
* Responsible for verifying signed licenses
*/
public class LicenseVerifier {

  /**
   * verifies the license content with the signature using the packaged
   * public key
   * @param license to verify
   * @return true if valid, false otherwise
   */
```

```
    public static boolean verifyLicense(final License license, byte[] encryptedPublicKeyData) {
        byte[] signedContent = null;
        byte[] signatureHash = null;
        try {
            byte[] signatureBytes = Base64.getDecoder().decode(license.signature());
            ByteBuffer byteBuffer = ByteBuffer.wrap(signatureBytes);
            int version = byteBuffer.getInt();
            int magicLen = byteBuffer.getInt();
            byte[] magic = new byte[magicLen];
            byteBuffer.get(magic);
            int hashLen = byteBuffer.getInt();
            signatureHash = new byte[hashLen];
            byteBuffer.get(signatureHash);
            int signedContentLen = byteBuffer.getInt();
            signedContent = new byte[signedContentLen];
            byteBuffer.get(signedContent);
            XContentBuilder contentBuilder = XContentFactory.contentBuilder(XContentType.JSON);
            license.toXContent(contentBuilder, new
ToXContent.MapParams(Collections.singletonMap(License.LICENSE_SPEC_VIEW_MODE, "true")));
            Signature rsa = Signature.getInstance("SHA512withRSA");
            rsa.initVerify(CryptUtils.readEncryptedPublicKey(encryptedPublicKeyData));
            BytesRefIterator iterator = BytesReference.bytes(contentBuilder).iterator();
            BytesRef ref;
            while((ref = iterator.next()) != null) {
                rsa.update(ref.bytes, ref.offset, ref.length);
            }
            return rsa.verify(signedContent)
                    && Arrays.equals(Base64.getEncoder().encode(encryptedPublicKeyData), signatureHash);
        } catch (IOException | NoSuchAlgorithmException | SignatureException | InvalidKeyException e) {
            throw new IllegalStateException(e);
        } finally {
            Arrays.fill(encryptedPublicKeyData, (byte) 0);
            if (signedContent != null) {
                Arrays.fill(signedContent, (byte) 0);
            }
            if (signatureHash != null) {
                Arrays.fill(signatureHash, (byte) 0);
            }
        }
    }

    public static boolean verifyLicense(final License license) {
        final byte[] publicKeyBytes;
        try (InputStream is = LicenseVerifier.class.getResourceAsStream("/public.key")) {
            ByteArrayOutputStream out = new ByteArrayOutputStream();
            Streams.copy(is, out);
            publicKeyBytes = out.toByteArray();
        } catch (IOException ex) {
```

```
      throw new IllegalStateException(ex);
    }
    return verifyLicense(license, publicKeyBytes);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.message.ParameterizedMessage;
import org.elasticsearch.Version;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.time.Clock;
import java.util.Collections;
import java.util.Map;
import java.util.UUID;
import java.util.concurrent.atomic.AtomicReference;

public class StartBasicClusterTask extends ClusterStateUpdateTask {

  private static final String ACKNOWLEDGEMENT_HEADER = "This license update requires acknowledgement.
To acknowledge the license, " +
      "please read the following messages and call /start_basic again, this time with the \"acknowledge=true\"
parameter:";

  private final Logger logger;
  private final String clusterName;
  private final PostStartBasicRequest request;
  private final ActionListener<PostStartBasicResponse> listener;
  private final Clock clock;
  private AtomicReference<Map<String, String[]>> ackMessages = new
AtomicReference<>(Collections.emptyMap());

  StartBasicClusterTask(Logger logger, String clusterName, Clock clock, PostStartBasicRequest request,
                ActionListener<PostStartBasicResponse> listener) {
    this.logger = logger;
    this.clusterName = clusterName;
    this.request = request;
    this.listener = listener;
    this.clock = clock;
  }

  @Override
  public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
    LicensesMetaData oldLicensesMetaData = oldState.metaData().custom(LicensesMetaData.TYPE);
    logger.debug("license prior to starting basic license: {}", oldLicensesMetaData);
    License oldLicense = LicensesMetaData.extractLicense(oldLicensesMetaData);
    Map<String, String[]> acknowledgeMessages = ackMessages.get();
    if (acknowledgeMessages.isEmpty() == false) {
      listener.onResponse(new
PostStartBasicResponse(PostStartBasicResponse.Status.NEED_ACKNOWLEDGEMENT, acknowledgeMessages,
          ACKNOWLEDGEMENT_HEADER));
    } else if (oldLicense != null && oldLicense.type().equals("basic")) {
      listener.onResponse(new
PostStartBasicResponse(PostStartBasicResponse.Status.ALREADY_USING_BASIC));
    }  else {
      listener.onResponse(new PostStartBasicResponse(PostStartBasicResponse.Status.GENERATED_BASIC));
    }
  }

  @Override
  public ClusterState execute(ClusterState currentState) throws Exception {
    XPackPlugin.checkReadyForXPackCustomMetadata(currentState);
    LicensesMetaData licensesMetaData = currentState.metaData().custom(LicensesMetaData.TYPE);
    License currentLicense = LicensesMetaData.extractLicense(licensesMetaData);
    if (currentLicense == null || currentLicense.type().equals("basic") == false) {
      long issueDate = clock.millis();
      MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
      License.Builder specBuilder = License.builder()
          .uid(UUID.randomUUID().toString())
          .issuedTo(clusterName)
          .maxNodes(LicenseService.SELF_GENERATED_LICENSE_MAX_NODES)
          .issueDate(issueDate)
          .type("basic")
          .expiryDate(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS);
      License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder);
      if (request.isAcknowledged() == false && currentLicense != null) {
        Map<String, String[]> ackMessages = LicenseService.getAckMessages(selfGeneratedLicense,
currentLicense);
        if (ackMessages.isEmpty() == false) {
          this.ackMessages.set(ackMessages);
          return currentState;
        }
      }
      Version trialVersion = null;
      if (licensesMetaData != null) {
```

```
                trialVersion = licensesMetaData.getMostRecentTrialVersion();
            }
            LicensesMetaData newLicensesMetaData = new LicensesMetaData(selfGeneratedLicense, trialVersion);
            mdBuilder.putCustom(LicensesMetaData.TYPE, newLicensesMetaData);
            return ClusterState.builder(currentState).metaData(mdBuilder).build();
        } else {
            return currentState;
        }
    }

    @Override
    public void onFailure(String source, @Nullable Exception e) {
        logger.error(new ParameterizedMessage("unexpected failure during [{}]", source), e);
        listener.onFailure(e);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.Strings;
import org.elasticsearch.common.logging.LoggerMessageFormat;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.XPackSettings;
import org.elasticsearch.xpack.core.monitoring.MonitoringField;

import java.util.Collections;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.concurrent.CopyOnWriteArrayList;
import java.util.function.BiFunction;

/**
* A holder for the current state of the license for all xpack features.
*/
public class XPackLicenseState {

    /** Messages for each feature which are printed when the license expires. */
    static final Map<String, String[]> EXPIRATION_MESSAGES;
    static {
        Map<String, String[]> messages = new LinkedHashMap<>();
        messages.put(XPackField.SECURITY, new String[] {
            "Cluster health, cluster stats and indices stats operations are blocked",
            "All data operations (read and write) continue to work"
        });
        messages.put(XPackField.WATCHER, new String[] {
            "PUT / GET watch APIs are disabled, DELETE watch API continues to work",
            "Watches execute and write to the history",
            "The actions of the watches don't execute"
        });
        messages.put(XPackField.MONITORING, new String[] {
            "The agent will stop collecting cluster and indices metrics",
            "The agent will stop automatically cleaning indices older than [xpack.monitoring.history.duration]"
        });
        messages.put(XPackField.GRAPH, new String[] {
            "Graph explore APIs are disabled"
        });
        messages.put(XPackField.MACHINE_LEARNING, new String[] {
            "Machine learning APIs are disabled"
        });
        messages.put(XPackField.LOGSTASH, new String[] {
            "Logstash will continue to poll centrally-managed pipelines"
        });
        messages.put(XPackField.DEPRECATION, new String[] {
            "Deprecation APIs are disabled"
        });
        messages.put(XPackField.UPGRADE, new String[] {
            "Upgrade API is disabled"
        });
        messages.put(XPackField.SQL, new String[] {
            "SQL support is disabled"
        });
        messages.put(XPackField.ROLLUP, new String[] {
            "Creating and Starting rollup jobs will no longer be allowed.",
            "Stopping/Deleting existing jobs, RollupCaps API and RollupSearch continue to function."
        });
        EXPIRATION_MESSAGES = Collections.unmodifiableMap(messages);
    }

    /**
     * Messages for each feature which are printed when the license type changes.
     * The value is a function taking the old and new license type, and returns the messages for that feature.
     */
    static final Map<String, BiFunction<OperationMode, OperationMode, String[]>>
ACKNOWLEDGMENT_MESSAGES;
    static {
        Map<String, BiFunction<OperationMode, OperationMode, String[]>> messages = new LinkedHashMap<>();
        messages.put(XPackField.SECURITY, XPackLicenseState::securityAcknowledgementMessages);
        messages.put(XPackField.WATCHER, XPackLicenseState::watcherAcknowledgementMessages);
        messages.put(XPackField.MONITORING, XPackLicenseState::monitoringAcknowledgementMessages);
        messages.put(XPackField.GRAPH, XPackLicenseState::graphAcknowledgementMessages);
        messages.put(XPackField.MACHINE_LEARNING,
XPackLicenseState::machineLearningAcknowledgementMessages);
        messages.put(XPackField.LOGSTASH, XPackLicenseState::logstashAcknowledgementMessages);
        messages.put(XPackField.SQL, XPackLicenseState::sqlAcknowledgementMessages);
        ACKNOWLEDGMENT_MESSAGES = Collections.unmodifiableMap(messages);
    }

    private static String[] securityAcknowledgementMessages(OperationMode currentMode, OperationMode
newMode) {
        switch (newMode) {
            case BASIC:
                switch (currentMode) {
                    case TRIAL:
                    case STANDARD:
                    case GOLD:
                    case PLATINUM:
                        return new String[] {
                            "The following X-Pack security functionality will be disabled: authentication, authorization, " +
                                "ip filtering, and auditing. Please restart your node after applying the license.",
                            "Field and document level access control will be disabled.",
                            "Custom realms will be ignored."
                        };
                }
                break;
            case GOLD:
                switch (currentMode) {
                    case BASIC:
                    case STANDARD:
                        // ^^ though technically it was already disabled, it's not bad to remind them
                    case TRIAL:
                    case PLATINUM:
                        return new String[] {
                            "Field and document level access control will be disabled.",
                            "Custom realms will be ignored."
                        };
                }
                break;
            case STANDARD:
                switch (currentMode) {
                    case BASIC:
                        // ^^ though technically it was already disabled, it's not bad to remind them
                    case GOLD:
                    case PLATINUM:
                    case TRIAL:
                        return new String[] {
                            "Authentication will be limited to the native realms.",
                        "IP filtering and auditing will be disabled.",
                        "Field and document level access control will be disabled.",
                        "Custom realms will be ignored."
                    };
                }
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] watcherAcknowledgementMessages(OperationMode currentMode, OperationMode
newMode) {
        switch (newMode) {
            case BASIC:
                switch (currentMode) {
                    case TRIAL:
                    case STANDARD:
                    case GOLD:
                    case PLATINUM:
                        return new String[] { "Watcher will be disabled" };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] monitoringAcknowledgementMessages(OperationMode currentMode, OperationMode
newMode) {
        switch (newMode) {
            case BASIC:
                switch (currentMode) {
                    case TRIAL:
                    case STANDARD:
                    case GOLD:
                    case PLATINUM:
                        return new String[] {
                            LoggerMessageFormat.format(
                                "Multi-cluster support is disabled for clusters with [{}] license. If you are\n" +
                                    "running multiple clusters, users won't be able to access the clusters with\n" +
                                    "[{}] licenses from within a single X-Pack Kibana instance. You will have to deploy a\n" +
                                    "separate and dedicated X-pack Kibana instance for each [{}] cluster you wish to monitor.",
                                newMode, newMode, newMode),
                            LoggerMessageFormat.format(
                                "Automatic index cleanup is locked to {} days for clusters with [{}] license.",
                                MonitoringField.HISTORY_DURATION.getDefault(Settings.EMPTY).days(), newMode)
                        };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] graphAcknowledgementMessages(OperationMode currentMode, OperationMode newMode)
{
        switch (newMode) {
            case BASIC:
            case STANDARD:
            case GOLD:
                switch (currentMode) {
                    case TRIAL:
                    case PLATINUM:
                        return new String[] { "Graph will be disabled" };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] machineLearningAcknowledgementMessages(OperationMode currentMode,
OperationMode newMode) {
        switch (newMode) {
            case BASIC:
            case STANDARD:
            case GOLD:
                switch (currentMode) {
                    case TRIAL:
                    case PLATINUM:
                        return new String[] { "Machine learning will be disabled" };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] logstashAcknowledgementMessages(OperationMode currentMode, OperationMode
newMode) {
        switch (newMode) {
            case BASIC:
                switch (currentMode) {
                    case TRIAL:
                    case STANDARD:
                    case GOLD:
                    case PLATINUM:
                        return new String[] { "Logstash will no longer poll for centrally-managed pipelines" };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] sqlAcknowledgementMessages(OperationMode currentMode, OperationMode newMode) {
        switch (newMode) {
            case BASIC:
            case STANDARD:
            case GOLD:
                switch (currentMode) {
                    case TRIAL:
                    case PLATINUM:
                        return new String[] { "JDBC support will be disabled, but you can continue to use SQL CLI and
REST endpoint" };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    /** A wrapper for the license mode and state, to allow atomically swapping. */
    private static class Status {

        /** The current "mode" of the license (ie license type). */
        final OperationMode mode;

        /** True if the license is active, or false if it is expired. */
        final boolean active;

        Status(OperationMode mode, boolean active) {
            this.mode = mode;
            this.active = active;
        }
    }

    private volatile Status status = new Status(OperationMode.TRIAL, true);
    private final List<Runnable> listeners = new CopyOnWriteArrayList<>();
    private final boolean isSecurityEnabled;
    private final boolean isSecurityExplicitlyEnabled;

    public XPackLicenseState(Settings settings) {
        this.isSecurityEnabled = XPackSettings.SECURITY_ENABLED.get(settings);
        // 6.0+ requires TLS for production licenses, so if TLS is enabled and security is enabled
        // we can interpret this as an explicit enabling of security if the security enabled
        // setting is not explicitly set
        this.isSecurityExplicitlyEnabled = isSecurityEnabled &&
            (settings.hasValue(XPackSettings.SECURITY_ENABLED.getKey()) ||
XPackSettings.TRANSPORT_SSL_ENABLED.get(settings));
    }
```

```
  /** Updates the current state of the license, which will change what features are available. */
  void update(OperationMode mode, boolean active) {
      status = new Status(mode, active);
      listeners.forEach(Runnable::run);
  }

  /** Add a listener to be notified on license change */
  public void addListener(Runnable runnable) {
      listeners.add(Objects.requireNonNull(runnable));
  }

  /** Remove a listener */
  public void removeListener(Runnable runnable) {
      listeners.remove(runnable);
  }

  /** Return the current license type. */
  public OperationMode getOperationMode() {
      return status.mode;
  }

  /** Return true if the license is currently within its time boundaries, false otherwise. */
  public boolean isActive() {
      return status.active;
  }

  /**
   * @return true if authentication and authorization should be enabled. this does not indicate what realms are
available
   * @see #allowedRealmType() for the enabled realms
   */
  public boolean isAuthAllowed() {
      OperationMode mode = status.mode;
      return mode == OperationMode.STANDARD || mode == OperationMode.GOLD || mode ==
OperationMode.PLATINUM
          || mode == OperationMode.TRIAL;
  }

  /**
   * @return true if IP filtering should be enabled
   */
  public boolean isIpFilteringAllowed() {
      OperationMode mode = status.mode;
      return mode == OperationMode.GOLD || mode == OperationMode.PLATINUM
              || mode == OperationMode.TRIAL;
  }
```

```
    /**
     * @return true if auditing should be enabled
     */
    public boolean isAuditingAllowed() {
        OperationMode mode = status.mode;
        return mode == OperationMode.GOLD || mode == OperationMode.PLATINUM
                || mode == OperationMode.TRIAL;
    }

    /**
     * Indicates whether the stats and health API calls should be allowed. If a license is expired and past the grace
     * period then we deny these calls.
     *
     * @return true if the license allows for the stats and health APIs to be used.
     */
    public boolean isStatsAndHealthAllowed() {
        return status.active;
    }

    /**
     * Determine if Document Level Security (DLS) and Field Level Security (FLS) should be enabled.
     * <p>
     * DLS and FLS are only disabled when the mode is not:
     * <ul>
     * <li>{@link OperationMode#PLATINUM}</li>
     * <li>{@link OperationMode#TRIAL}</li>
     * </ul>
     * Note: This does not consider the <em>state</em> of the license so that Security does not suddenly leak
information!
     *
     * @return {@code true} to enable DLS and FLS. Otherwise {@code false}.
     */
    public boolean isDocumentAndFieldLevelSecurityAllowed() {
        OperationMode mode = status.mode;
        return mode == OperationMode.TRIAL || mode == OperationMode.PLATINUM;
    }

    /** Classes of realms that may be available based on the license type. */
    public enum AllowedRealmType {
        NONE,
        NATIVE,
        DEFAULT,
        ALL
    }

    /**
     * @return the type of realms that are enabled based on the license {@link OperationMode}
     */
```

```
public AllowedRealmType allowedRealmType() {
    switch (status.mode) {
        case PLATINUM:
        case TRIAL:
            return AllowedRealmType.ALL;
        case GOLD:
            return AllowedRealmType.DEFAULT;
        case STANDARD:
            return AllowedRealmType.NATIVE;
        default:
            return AllowedRealmType.NONE;
    }
}

/**
 * @return whether custom role providers are allowed based on the license {@link OperationMode}
 */
public boolean isCustomRoleProvidersAllowed() {
    final Status localStatus = status;
    return (localStatus.mode == OperationMode.PLATINUM || localStatus.mode == OperationMode.TRIAL )
            && localStatus.active;
}

/**
 * Determine if Watcher is available based on the current license.
 * <p>
 * Watcher is available if the license is active (hasn't expired) and of one of the following types:
 * <ul>
 * <li>{@link OperationMode#STANDARD}</li>
 * <li>{@link OperationMode#PLATINUM}</li>
 * <li>{@link OperationMode#GOLD}</li>
 * <li>{@link OperationMode#TRIAL}</li>
 * </ul>
 *
 * @return {@code true} as long as the license is valid. Otherwise {@code false}.
 */
public boolean isWatcherAllowed() {
    // status is volatile, so a local variable is used for a consistent view
    Status localStatus = status;

    if (localStatus.active == false) {
        return false;
    }

    switch (localStatus.mode) {
        case TRIAL:
        case GOLD:
        case PLATINUM:
            case STANDARD:
                return true;
            default:
                return false;
        }
    }

    /**
     * Monitoring is always available as long as there is a valid license
     *
     * @return true if the license is active
     */
    public boolean isMonitoringAllowed() {
        return status.active;
    }

    /**
     * Monitoring Cluster Alerts requires the equivalent license to use Watcher.
     *
     * @return {@link #isWatcherAllowed()}
     * @see #isWatcherAllowed()
     */
    public boolean isMonitoringClusterAlertsAllowed() {
        return isWatcherAllowed();
    }

    /**
     * Determine if the current license allows the retention of indices to be modified.
     * <p>
     * Only users with a non-{@link OperationMode#BASIC} license can update the retention period.
     * <p>
     * Note: This does not consider the <em>state</em> of the license so that any change is remembered for when
they fix their license.
     *
     * @return {@code true} if the user is allowed to modify the retention. Otherwise {@code false}.
     */
    public boolean isUpdateRetentionAllowed() {
        final OperationMode mode = status.mode;
        return mode != OperationMode.BASIC && mode != OperationMode.MISSING;
    }

    /**
     * Determine if Graph Exploration should be enabled.
     * <p>
     * Exploration is only disabled when the license has expired or if the mode is not:
     * <ul>
     * <li>{@link OperationMode#PLATINUM}</li>
     * <li>{@link OperationMode#TRIAL}</li>
     * </ul>
     *
     * @return {@code true} as long as the license is valid. Otherwise {@code false}.
     */
    public boolean isGraphAllowed() {
        // status is volatile
        Status localStatus = status;
        OperationMode operationMode = localStatus.mode;

        boolean licensed = operationMode == OperationMode.TRIAL || operationMode ==
OperationMode.PLATINUM;

        return licensed && localStatus.active;
    }

    /**
     * Determine if Machine Learning should be enabled.
     * <p>
     * Machine Learning is only disabled when the license has expired or if the
     * mode is not:
     * <ul>
     * <li>{@link OperationMode#PLATINUM}</li>
     * <li>{@link OperationMode#TRIAL}</li>
     * </ul>
     *
     * @return {@code true} as long as the license is valid. Otherwise
     *         {@code false}.
     */
    public boolean isMachineLearningAllowed() {
        // status is volatile
        Status localStatus = status;
        OperationMode operationMode = localStatus.mode;

        boolean licensed = operationMode == OperationMode.TRIAL || operationMode ==
OperationMode.PLATINUM;

        return licensed && localStatus.active;
    }

    /**
     * Rollup is always available as long as there is a valid license
     *
     * @return true if the license is active
     */
    public boolean isRollupAllowed() {
        return status.active;
    }
```

```
/**
 * Logstash is allowed as long as there is an active license of type TRIAL, STANDARD, GOLD or PLATINUM
 * @return {@code true} as long as there is a valid license
 */
public boolean isLogstashAllowed() {
    Status localStatus = status;

    if (localStatus.active == false) {
        return false;
    }

    switch (localStatus.mode) {
        case TRIAL:
        case GOLD:
        case PLATINUM:
        case STANDARD:
            return true;
        default:
            return false;
    }
}

/**
 * Deprecation APIs are always allowed as long as there is an active license
 * @return {@code true} as long as there is a valid license
 */
public boolean isDeprecationAllowed() {
    return status.active;
}

/**
 * Determine if Upgrade API should be enabled.
 * <p>
 *  Upgrade API is not available in for all license types except {@link OperationMode#MISSING}
 *
 * @return {@code true} as long as the license is valid. Otherwise
 *         {@code false}.
 */
public boolean isUpgradeAllowed() {
    // status is volatile
    Status localStatus = status;
    // Should work on all active licenses
    return localStatus.active;
}

/**
 * Determine if SQL support should be enabled.
 * <p>
```

```
     *  SQL is available for all license types except {@link OperationMode#MISSING}
     */
    public boolean isSqlAllowed() {
        return status.active;
    }

    /**
     * Determine if JDBC support should be enabled.
     * <p>
     *  JDBC is available only in for {@link OperationMode#PLATINUM} and {@link OperationMode#TRIAL}
licences
     */
    public boolean isJdbcAllowed() {
        // status is volatile
        Status localStatus = status;
        OperationMode operationMode = localStatus.mode;

        boolean licensed = operationMode == OperationMode.TRIAL || operationMode ==
OperationMode.PLATINUM;

        return licensed && localStatus.active;
    }

    public boolean isTrialLicense() {
        return status.mode == OperationMode.TRIAL;
    }

    public boolean isSecurityAvailable() {
        OperationMode mode = status.mode;
        return mode == OperationMode.GOLD || mode == OperationMode.PLATINUM || mode ==
OperationMode.STANDARD ||
                mode == OperationMode.TRIAL;
    }

    public boolean isSecurityEnabled() {
        final OperationMode mode = status.mode;
        return mode == OperationMode.TRIAL ? isSecurityExplicitlyEnabled : isSecurityEnabled;
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.AcknowledgedResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

public class DeleteLicenseResponse extends AcknowledgedResponse {

    DeleteLicenseResponse() {
    }

    DeleteLicenseResponse(boolean acknowledged) {
        super(acknowledged);
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        super.readFrom(in);
        readAcknowledged(in);
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        super.writeTo(out);
        writeAcknowledged(out);
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.AcknowledgedResponse;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.XContentBuilder;

import java.io.IOException;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

public class PutLicenseResponse extends AcknowledgedResponse {

    private LicensesStatus status;
    private Map<String, String[]> acknowledgeMessages;
    private String acknowledgeHeader;

    PutLicenseResponse() {
    }

    public PutLicenseResponse(boolean acknowledged, LicensesStatus status) {
        this(acknowledged, status, null, Collections.<String, String[]>emptyMap());
    }

    public PutLicenseResponse(boolean acknowledged, LicensesStatus status, String acknowledgeHeader,
                              Map<String, String[]> acknowledgeMessages) {
        super(acknowledged);
        this.status = status;
        this.acknowledgeHeader = acknowledgeHeader;
        this.acknowledgeMessages = acknowledgeMessages;
    }

    public LicensesStatus status() {
        return status;
    }

    public Map<String, String[]> acknowledgeMessages() {
        return acknowledgeMessages;
    }

    public String acknowledgeHeader() {
        return acknowledgeHeader;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        super.readFrom(in);
        readAcknowledged(in);
        status = LicensesStatus.fromId(in.readVInt());
        acknowledgeHeader = in.readOptionalString();
        int size = in.readVInt();
        Map<String, String[]> acknowledgeMessages = new HashMap<>(size);
        for (int i = 0; i < size; i++) {
            String feature = in.readString();
            int nMessages = in.readVInt();
            String[] messages = new String[nMessages];
            for (int j = 0; j < nMessages; j++) {
                messages[j] = in.readString();
            }
            acknowledgeMessages.put(feature, messages);
        }
        this.acknowledgeMessages = acknowledgeMessages;
    }
```

```
@Override
public void writeTo(StreamOutput out) throws IOException {
    super.writeTo(out);
    writeAcknowledged(out);
    out.writeVInt(status.id());
    out.writeOptionalString(acknowledgeHeader);
    out.writeVInt(acknowledgeMessages.size());
    for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
        out.writeString(entry.getKey());
        out.writeVInt(entry.getValue().length);
        for (String message : entry.getValue()) {
            out.writeString(message);
        }
    }
}

@Override
protected void addCustomFields(XContentBuilder builder, Params params) throws IOException {
    switch (status) {
        case VALID:
            builder.field("license_status", "valid");
            break;
        case INVALID:
            builder.field("license_status", "invalid");
            break;
        case EXPIRED:
            builder.field("license_status", "expired");
            break;
        default:
            throw new IllegalArgumentException("unknown status [" + status + "] found");
    }
    if (!acknowledgeMessages.isEmpty()) {
        builder.startObject("acknowledge");
        builder.field("message", acknowledgeHeader);
        for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
            builder.startArray(entry.getKey());
            for (String message : entry.getValue()) {
                builder.value(message);
            }
            builder.endArray();
        }
        builder.endObject();
    }
}

@Override
public String toString() {
        return Strings.toString(this, true, true);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

class PostStartTrialRequestBuilder extends ActionRequestBuilder<PostStartTrialRequest,
        PostStartTrialResponse, PostStartTrialRequestBuilder> {

    PostStartTrialRequestBuilder(ElasticsearchClient client, PostStartTrialAction action) {
        super(client, action, new PostStartTrialRequest());
    }

    public PostStartTrialRequestBuilder setAcknowledge(boolean acknowledge) {
        request.acknowledge(acknowledge);
        return this;
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeAction;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ack.ClusterStateUpdateResponse;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportDeleteLicenseAction extends TransportMasterNodeAction<DeleteLicenseRequest,
DeleteLicenseResponse> {

    private final LicenseService licenseService;

    @Inject
    public TransportDeleteLicenseAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                       LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                                       IndexNameExpressionResolver indexNameExpressionResolver) {
        super(settings, DeleteLicenseAction.NAME, transportService, clusterService, threadPool, actionFilters,
                indexNameExpressionResolver, DeleteLicenseRequest::new);
        this.licenseService = licenseService;
    }

    @Override
    protected String executor() {
        return ThreadPool.Names.MANAGEMENT;
    }

    @Override
    protected DeleteLicenseResponse newResponse() {
        return new DeleteLicenseResponse();
    }

    @Override
    protected ClusterBlockException checkBlock(DeleteLicenseRequest request, ClusterState state) {
        return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_WRITE);
    }

    @Override
    protected void masterOperation(final DeleteLicenseRequest request, ClusterState state, final
ActionListener<DeleteLicenseResponse>
           listener) throws ElasticsearchException {
        licenseService.removeLicense(request, new ActionListener<ClusterStateUpdateResponse>() {
           @Override
           public void onResponse(ClusterStateUpdateResponse clusterStateUpdateResponse) {
                listener.onResponse(new DeleteLicenseResponse(clusterStateUpdateResponse.isAcknowledged()));
           }

           @Override
           public void onFailure(Exception e) {
                listener.onFailure(e);
           }
        });
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;


import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.message.ParameterizedMessage;
import org.apache.logging.log4j.util.Supplier;
import org.apache.lucene.util.BytesRef;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.watcher.FileChangesListener;
import org.elasticsearch.watcher.FileWatcher;
import org.elasticsearch.watcher.ResourceWatcherService;

import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.concurrent.atomic.AtomicBoolean;

/**
* File based watcher for license {@link OperationMode}
* Watches for changes in <code>licenseModePath</code>, use
* {@link #getCurrentOperationMode()} to access the latest mode
*
* In case of failure to read a valid operation mode from <code>licenseModePath</code>,
* the operation mode will default to PLATINUM
*/
public final class OperationModeFileWatcher implements FileChangesListener {
  private final ResourceWatcherService resourceWatcherService;
  private final Path licenseModePath;
  private final AtomicBoolean initialized = new AtomicBoolean();
  private final OperationMode defaultOperationMode = OperationMode.PLATINUM;
  private volatile OperationMode currentOperationMode = defaultOperationMode;
  private final Logger logger;
  private final Runnable onChange;

  public OperationModeFileWatcher(ResourceWatcherService resourceWatcherService, Path licenseModePath,
                                  Logger logger, Runnable onChange) {
    this.resourceWatcherService = resourceWatcherService;
    this.licenseModePath = licenseModePath;
    this.logger = logger;
    this.onChange = onChange;
  }

  public void init() {
    if (initialized.compareAndSet(false, true)) {
        final FileWatcher watcher = new FileWatcher(licenseModePath);
        watcher.addListener(this);
        try {
            resourceWatcherService.add(watcher, ResourceWatcherService.Frequency.HIGH);
            if (Files.exists(licenseModePath)) {
                onChange(licenseModePath);
            }
        } catch (IOException e) {
            logger.error("couldn't initialize watching license mode file", e);
        }
    }
}

/**
 * Returns the current operation mode based on license mode file.
 * Defaults to {@link OperationMode#PLATINUM}
 */
public OperationMode getCurrentOperationMode() {
    return currentOperationMode;
}

@Override
public void onFileInit(Path file) {
    onChange(file);
}

@Override
public void onFileCreated(Path file) {
    onChange(file);
}

@Override
public void onFileDeleted(Path file) {
    onChange(file);
}

@Override
public void onFileChanged(Path file) {
    onChange(file);
}

private synchronized void onChange(Path file) {
    if (file.equals(licenseModePath)) {
        OperationMode newOperationMode = defaultOperationMode;
        try {
            if (Files.exists(licenseModePath)
                    && Files.isReadable(licenseModePath)) {
```

```
                    final byte[] content;
                    try {
                        content = Files.readAllBytes(licenseModePath);
                    } catch (IOException e) {
                        logger.error(
                                (Supplier<?>) () -> new ParameterizedMessage(
                                        "couldn't read operation mode from [{}]", licenseModePath.toAbsolutePath()), e);
                        return;
                    }
                    // this UTF-8 conversion is much pickier than java String
                    final String operationMode = new BytesRef(content).utf8ToString();
                    try {
                        newOperationMode = OperationMode.resolve(operationMode);
                    } catch (IllegalArgumentException e) {
                        logger.error(
                                (Supplier<?>) () -> new ParameterizedMessage(
                                        "invalid operation mode in [{}]", licenseModePath.toAbsolutePath()), e);
                        return;
                    }
                }
            } finally {
                // set this after the fact to prevent that we are jumping back and forth first setting to defautl and then
reading the
                // actual op mode resetting it.
                this.currentOperationMode = newOperationMode;
            }
            onChange.run();
        }
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

class GetTrialStatusRequestBuilder extends ActionRequestBuilder<GetTrialStatusRequest,
        GetTrialStatusResponse, GetTrialStatusRequestBuilder> {

    GetTrialStatusRequestBuilder(ElasticsearchClient client, GetTrialStatusAction action) {
        super(client, action, new GetTrialStatusRequest());
    }
}
```

```
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.ValidateActions;
import org.elasticsearch.action.support.master.AcknowledgedRequest;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.XContentType;

import java.io.IOException;


public class PutLicenseRequest extends AcknowledgedRequest<PutLicenseRequest> {

    private License license;
    private boolean acknowledge = false;

    public PutLicenseRequest() {
    }

    @Override
    public ActionRequestValidationException validate() {
        return (license == null) ? ValidateActions.addValidationError("license is missing", null) : null;
    }

    /**
     * Parses license from json format to an instance of {@link License}
     *
     * @param licenseDefinition licenses definition
     * @param xContentType the content type of the license
     */
    public PutLicenseRequest license(BytesReference licenseDefinition, XContentType xContentType) {
        try {
            return license(License.fromSource(licenseDefinition, xContentType));
        } catch (IOException e) {
            throw new IllegalArgumentException("failed to parse license source", e);
        }
    }

    public PutLicenseRequest license(License license) {
        this.license = license;
        return this;
```

```
  }

  public License license() {
    return license;
  }

  public PutLicenseRequest acknowledge(boolean acknowledge) {
    this.acknowledge = acknowledge;
    return this;
  }

  public boolean acknowledged() {
    return acknowledge;
  }

  @Override
  public void readFrom(StreamInput in) throws IOException {
    super.readFrom(in);
    license = License.readLicense(in);
    acknowledge = in.readBoolean();
  }

  @Override
  public void writeTo(StreamOutput out) throws IOException {
    super.writeTo(out);
    license.writeTo(out);
    out.writeBoolean(acknowledge);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.LoggingDeprecationHandler;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.common.xcontent.XContentType;

import java.io.IOException;
import java.nio.ByteBuffer;
import java.util.Base64;
import java.util.Collections;

import static org.elasticsearch.license.CryptUtils.decrypt;
import static org.elasticsearch.license.CryptUtils.encrypt;

class SelfGeneratedLicense {

  public static License create(License.Builder specBuilder) {
    License spec = specBuilder
        .issuer("elasticsearch")
        .version(License.VERSION_CURRENT)
        .build();
    final String signature;
    try {
      XContentBuilder contentBuilder = XContentFactory.contentBuilder(XContentType.JSON);
      spec.toXContent(contentBuilder, new
ToXContent.MapParams(Collections.singletonMap(License.LICENSE_SPEC_VIEW_MODE, "true")));
      byte[] encrypt = encrypt(BytesReference.toBytes(BytesReference.bytes(contentBuilder)));
      byte[] bytes = new byte[4 + 4 + encrypt.length];
      ByteBuffer byteBuffer = ByteBuffer.wrap(bytes);
      // always generate license version -VERSION_CURRENT
      byteBuffer.putInt(-License.VERSION_CURRENT)
          .putInt(encrypt.length)
          .put(encrypt);
      signature = Base64.getEncoder().encodeToString(bytes);
    } catch (IOException e) {
      throw new IllegalStateException(e);
    }
    return License.builder().fromLicenseSpec(spec, signature).build();
  }

  public static boolean verify(final License license) {
    try {
      byte[] signatureBytes = Base64.getDecoder().decode(license.signature());
      ByteBuffer byteBuffer = ByteBuffer.wrap(signatureBytes);
      int version = byteBuffer.getInt();
      int contentLen = byteBuffer.getInt();
      byte[] content = new byte[contentLen];
      byteBuffer.get(content);
      final License expectedLicense;
      // EMPTY is safe here because we don't call namedObject
      try (XContentParser parser = XContentFactory.xContent(XContentType.JSON)
          .createParser(NamedXContentRegistry.EMPTY, LoggingDeprecationHandler.INSTANCE,
decrypt(content))) {
        parser.nextToken();
        expectedLicense = License.builder().fromLicenseSpec(License.fromXContent(parser),
            license.signature()).version(-version).build();
      }
```

```
            return license.equals(expectedLicense);
        } catch (IOException e) {
            throw new IllegalStateException(e);
        }
    }

    public static boolean validSelfGeneratedType(String type) {
        return "basic".equals(type) || "trial".equals(type);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;
import java.util.Map;

import static org.elasticsearch.rest.RestRequest.Method.POST;

public class RestPostStartTrialLicense extends XPackRestHandler {

    RestPostStartTrialLicense(Settings settings, RestController controller) {
        super(settings);
        controller.registerHandler(POST, URI_BASE + "/license/start_trial", this);
    }

    @Override
    protected RestChannelConsumer doPrepareRequest(RestRequest request, XPackClient client) throws IOException
{
        PostStartTrialRequest startTrialRequest = new PostStartTrialRequest();
        startTrialRequest.setType(request.param("type", "trial"));
        startTrialRequest.acknowledge(request.paramAsBoolean("acknowledge", false));
        return channel -> client.licensing().postStartTrial(startTrialRequest,
                new RestBuilderListener<PostStartTrialResponse>(channel) {
                    @Override
```

```
          public RestResponse buildResponse(PostStartTrialResponse response, XContentBuilder builder) throws
Exception {
            PostStartTrialResponse.Status status = response.getStatus();
            builder.startObject();
            builder.field("acknowledged", startTrialRequest.isAcknowledged());
            if (status.isTrialStarted()) {
              builder.field("trial_was_started", true);
              builder.field("type", startTrialRequest.getType());
            } else {
              builder.field("trial_was_started", false);
              builder.field("error_message", status.getErrorMessage());
            }

            Map<String, String[]> acknowledgementMessages = response.getAcknowledgementMessages();
            if (acknowledgementMessages.isEmpty() == false) {
              builder.startObject("acknowledge");
              builder.field("message", response.getAcknowledgementMessage());
              for (Map.Entry<String, String[]> entry : acknowledgementMessages.entrySet()) {
                builder.startArray(entry.getKey());
                for (String message : entry.getValue()) {
                  builder.value(message);
                }
                builder.endArray();
              }
              builder.endObject();
            }
            builder.endObject();
            return new BytesRestResponse(status.getRestStatus(), builder);
          }
        });
  }

  @Override
  public String getName() {
    return "xpack_upgrade_to_trial_action";
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeReadAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportGetLicenseAction extends TransportMasterNodeReadAction<GetLicenseRequest,
GetLicenseResponse> {

   private final LicenseService licenseService;

   @Inject
   public TransportGetLicenseAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                    LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                                    IndexNameExpressionResolver indexNameExpressionResolver) {
       super(settings, GetLicenseAction.NAME, transportService, clusterService, threadPool, actionFilters,
indexNameExpressionResolver,
               GetLicenseRequest::new);
       this.licenseService = licenseService;
   }

   @Override
   protected String executor() {
       return ThreadPool.Names.MANAGEMENT;
   }

   @Override
   protected GetLicenseResponse newResponse() {
       return new GetLicenseResponse();
   }

   @Override
   protected ClusterBlockException checkBlock(GetLicenseRequest request, ClusterState state) {
       return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_READ);
   }

   @Override
   protected void masterOperation(final GetLicenseRequest request, ClusterState state,
                                  final ActionListener<GetLicenseResponse> listener) throws ElasticsearchException {
       listener.onResponse(new GetLicenseResponse(licenseService.getLicense()));
   }
}
```

```
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.POST;

public class RestPostStartBasicLicense extends XPackRestHandler {

  RestPostStartBasicLicense(Settings settings, RestController controller) {
    super(settings);
    controller.registerHandler(POST, URI_BASE + "/license/start_basic", this);
  }

  @Override
  protected RestChannelConsumer doPrepareRequest(RestRequest request, XPackClient client) throws IOException
{
    PostStartBasicRequest startBasicRequest = new PostStartBasicRequest();
    startBasicRequest.acknowledge(request.paramAsBoolean("acknowledge", false));
    startBasicRequest.timeout(request.paramAsTime("timeout", startBasicRequest.timeout()));
    startBasicRequest.masterNodeTimeout(request.paramAsTime("master_timeout",
startBasicRequest.masterNodeTimeout()));
    return channel -> client.licensing().postStartBasic(startBasicRequest, new
RestBuilderListener<PostStartBasicResponse>(channel) {
      @Override
      public RestResponse buildResponse(PostStartBasicResponse response, XContentBuilder builder) throws
Exception {
        PostStartBasicResponse.Status status = response.getStatus();
        response.toXContent(builder, ToXContent.EMPTY_PARAMS);
        return new BytesRestResponse(status.getRestStatus(), builder);
      }
    });
  }
```

```
    @Override
    public String getName() {
        return "xpack_start_basic_action";
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.util.ArrayList;
import java.util.Base64;
import java.util.Comparator;
import java.util.List;
import java.util.Locale;

import org.apache.lucene.util.CollectionUtil;
import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.ElasticsearchParseException;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.LoggingDeprecationHandler;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.ToXContentObject;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.common.xcontent.XContentType;

/**
* Data structure for license. Use {@link Builder} to build a license.
* Provides serialization/deserialization & validation methods for license object
*/
public class License implements ToXContentObject {
    public static final int VERSION_START = 1;
    public static final int VERSION_NO_FEATURE_TYPE = 2;
    public static final int VERSION_START_DATE = 3;
    public static final int VERSION_CURRENT = VERSION_START_DATE;
```

```
  /**
   * XContent param name to deserialize license(s) with
   * an additional <code>status</code> field, indicating whether a
   * particular license is 'active' or 'expired' and no signature
   * and in a human readable format
   */
  public static final String REST_VIEW_MODE = "rest_view";
  /**
   * XContent param name to deserialize license(s) with
   * no signature
   */
  public static final String LICENSE_SPEC_VIEW_MODE = "license_spec_view";
  /**
   * XContent param name to deserialize licenses according
   * to a specific license version
   */
  public static final String LICENSE_VERSION_MODE = "license_version";

  public static final Comparator<License> LATEST_ISSUE_DATE_FIRST =
Comparator.comparing(License::issueDate).reversed();

  private final int version;
  private final String uid;
  private final String issuer;
  private final String issuedTo;
  private final long issueDate;
  private final String type;
  private final String subscriptionType;
  private final String feature;
  private final String signature;
  private final long expiryDate;
  private final long startDate;
  private final int maxNodes;
  private final OperationMode operationMode;

  /**
   * Decouples operation mode of a license from the license type value.
   * <p>
   * Note: The mode indicates features that should be made available, but it does not indicate whether the license is
active!
   *
   * The id byte is used for ordering operation modes (used for merging license md in tribe node)
   */
  public enum OperationMode {
    MISSING((byte) 0),
    TRIAL((byte) 1),
    BASIC((byte) 2),
    STANDARD((byte) 3),
    GOLD((byte) 4),
    PLATINUM((byte) 5);

    private final byte id;

    OperationMode(byte id) {
        this.id = id;
    }

    /** Returns non-zero positive number when <code>opMode1</code> is greater than <code>opMode2</code>
*/
    public static int compare(OperationMode opMode1, OperationMode opMode2) {
        return Integer.compare(opMode1.id, opMode2.id);
    }

    public static OperationMode resolve(String type) {
        switch (type.toLowerCase(Locale.ROOT)) {
            case "missing":
                return MISSING;
            case "trial":
            case "none": // bwc for 1.x subscription_type field
            case "dev": // bwc for 1.x subscription_type field
            case "development": // bwc for 1.x subscription_type field
                return TRIAL;
            case "basic":
                return BASIC;
            case "standard":
                return STANDARD;
            case "silver":
            case "gold":
                return GOLD;
            case "platinum":
            case "cloud_internal":
            case "internal": // bwc for 1.x subscription_type field
                return PLATINUM;
            default:
                throw new IllegalArgumentException("unknown type [" + type + "]");
        }
    }
  }

  private License(int version, String uid, String issuer, String issuedTo, long issueDate, String type,
                String subscriptionType, String feature, String signature, long expiryDate, int maxNodes, long startDate)
{
    this.version = version;
    this.uid = uid;
    this.issuer = issuer;
    this.issuedTo = issuedTo;
    this.issueDate = issueDate;
    this.type = type;
    this.subscriptionType = subscriptionType;
    this.feature = feature;
    this.signature = signature;
    // We will validate that only a basic license can have the
BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS
    // in the validate() method.
    if (expiryDate == -1) {
        this.expiryDate = LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
    } else {
        this.expiryDate = expiryDate;
    }
    this.maxNodes = maxNodes;
    this.startDate = startDate;
    if (version == VERSION_START) {
        // in 1.x: the acceptable values for 'subscription_type': none | dev | silver | gold | platinum
        this.operationMode = OperationMode.resolve(subscriptionType);
    } else {
        // in 2.x: the acceptable values for 'type': trial | basic | silver | dev | gold | platinum
        this.operationMode = OperationMode.resolve(type);
    }
    validate();
  }

  /**
   * @return version of the license
   */
  public int version() {
    return version;
  }

  /**
   * @return a unique identifier for a license
   */
  public String uid() {
    return uid;
  }

  /**
   * @return type of the license [trial, subscription, internal]
   */
  public String type() {
    return type;
  }

  /**
   * @return the issueDate in milliseconds
```

```
 */
public long issueDate() {
   return issueDate;
}

/**
 * @return the startDate in milliseconds
 */
public long startDate() {
   return startDate;
}

/**
 * @return the expiry date in milliseconds
 */
public long expiryDate() {
   return expiryDate;
}

/**
 * @return the maximum number of nodes this license has been issued for
 */
public int maxNodes() {
   return maxNodes;
}

/**
 * @return a string representing the entity this licenses has been issued to
 */
public String issuedTo() {
   return issuedTo;
}

/**
 * @return a string representing the entity responsible for issuing this license (internal)
 */
public String issuer() {
   return issuer;
}

/**
 * @return a string representing the signature of the license used for license verification
 */
public String signature() {
   return signature;
}

/**
```

```
     * @return the operation mode of the license as computed from the license type or from
     * the license mode file
     */
    public OperationMode operationMode() {
        synchronized (this) {
            if (canReadOperationModeFromFile() && operationModeFileWatcher != null) {
                return operationModeFileWatcher.getCurrentOperationMode();
            }
        }
        return operationMode;
    }

    private boolean canReadOperationModeFromFile() {
        return type.equals("cloud_internal");
    }

    private volatile OperationModeFileWatcher operationModeFileWatcher;

    /**
     * Sets the operation mode file watcher for the license and initializes the
     * file watcher when the license type allows to override operation mode from file
     */
    public synchronized void setOperationModeFileWatcher(final OperationModeFileWatcher
operationModeFileWatcher) {
        this.operationModeFileWatcher = operationModeFileWatcher;
        if (canReadOperationModeFromFile()) {
            this.operationModeFileWatcher.init();
        }
    }

    /**
     * Removes operation mode file watcher, so unused license objects can be gc'ed
     */
    public synchronized void removeOperationModeFileWatcher() {
        this.operationModeFileWatcher = null;
    }

    /**
     * @return the current license's status
     */
    public Status status() {
        long now = System.currentTimeMillis();
        if (issueDate > now) {
            return Status.INVALID;
        } else if (expiryDate < now) {
            return Status.EXPIRED;
        }
        return Status.ACTIVE;
    }

    private void validate() {
        if (issuer == null) {
            throw new IllegalStateException("issuer can not be null");
        } else if (issuedTo == null) {
            throw new IllegalStateException("issuedTo can not be null");
        } else if (issueDate == -1) {
            throw new IllegalStateException("issueDate has to be set");
        } else if (type == null) {
            throw new IllegalStateException("type can not be null");
        } else if (subscriptionType == null && version == VERSION_START) {
            throw new IllegalStateException("subscriptionType can not be null");
        } else if (uid == null) {
            throw new IllegalStateException("uid can not be null");
        } else if (feature == null && version == VERSION_START) {
            throw new IllegalStateException("feature can not be null");
        } else if (maxNodes == -1) {
            throw new IllegalStateException("maxNodes has to be set");
        } else if (expiryDate == -1) {
            throw new IllegalStateException("expiryDate has to be set");
        } else if (expiryDate == LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS
&& "basic".equals(type) == false) {
            throw new IllegalStateException("only basic licenses are allowed to have no expiration");
        }
    }

    public static License readLicense(StreamInput in) throws IOException {
        int version = in.readVInt(); // Version for future extensibility
        if (version > VERSION_CURRENT) {
            throw new ElasticsearchException("Unknown license version found, please upgrade all nodes to the latest
elasticsearch-license" +
                    " plugin");
        }
        Builder builder = builder();
        builder.version(version);
        builder.uid(in.readString());
        builder.type(in.readString());
        if (version == VERSION_START) {
            builder.subscriptionType(in.readString());
        }
        builder.issueDate(in.readLong());
        if (version == VERSION_START) {
            builder.feature(in.readString());
        }
        builder.expiryDate(in.readLong());
        builder.maxNodes(in.readInt());
        builder.issuedTo(in.readString());
        builder.issuer(in.readString());
        builder.signature(in.readOptionalString());
        if (version >= VERSION_START_DATE) {
            builder.startDate(in.readLong());
        }
        return builder.build();
    }

    public void writeTo(StreamOutput out) throws IOException {
        out.writeVInt(version);
        out.writeString(uid);
        out.writeString(type);
        if (version == VERSION_START) {
            out.writeString(subscriptionType);
        }
        out.writeLong(issueDate);
        if (version == VERSION_START) {
            out.writeString(feature);
        }
        out.writeLong(expiryDate);
        out.writeInt(maxNodes);
        out.writeString(issuedTo);
        out.writeString(issuer);
        out.writeOptionalString(signature);
        if (version >= VERSION_START_DATE) {
            out.writeLong(startDate);
        }
    }

    @Override
    public String toString() {
        try {
            final XContentBuilder builder = XContentFactory.jsonBuilder();
            toXContent(builder, ToXContent.EMPTY_PARAMS);
            return Strings.toString(builder);
        } catch (IOException e) {
            return "";
        }
    }

    @Override
    public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
        builder.startObject();
        toInnerXContent(builder, params);
        builder.endObject();
        return builder;
    }
    public XContentBuilder toInnerXContent(XContentBuilder builder, Params params) throws IOException {
        boolean licenseSpecMode = params.paramAsBoolean(LICENSE_SPEC_VIEW_MODE, false);
        boolean restViewMode = params.paramAsBoolean(REST_VIEW_MODE, false);
        boolean previouslyHumanReadable = builder.humanReadable();
        if (licenseSpecMode && restViewMode) {
            throw new IllegalArgumentException("can have either " + REST_VIEW_MODE + " or " +
LICENSE_SPEC_VIEW_MODE);
        } else if (restViewMode) {
            if (!previouslyHumanReadable) {
                builder.humanReadable(true);
            }
        }
        final int version;
        if (params.param(LICENSE_VERSION_MODE) != null && restViewMode) {
            version = Integer.parseInt(params.param(LICENSE_VERSION_MODE));
        } else {
            version = this.version;
        }
        if (restViewMode) {
            builder.field(Fields.STATUS, status().label());
        }
        builder.field(Fields.UID, uid);
        builder.field(Fields.TYPE, type);
        if (version == VERSION_START) {
            builder.field(Fields.SUBSCRIPTION_TYPE, subscriptionType);
        }
        builder.timeField(Fields.ISSUE_DATE_IN_MILLIS, Fields.ISSUE_DATE, issueDate);
        if (version == VERSION_START) {
            builder.field(Fields.FEATURE, feature);
        }

        if (expiryDate != LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS) {
            builder.timeField(Fields.EXPIRY_DATE_IN_MILLIS, Fields.EXPIRY_DATE, expiryDate);
        }
        builder.field(Fields.MAX_NODES, maxNodes);
        builder.field(Fields.ISSUED_TO, issuedTo);
        builder.field(Fields.ISSUER, issuer);
        if (!licenseSpecMode && !restViewMode && signature != null) {
            builder.field(Fields.SIGNATURE, signature);
        }
        if (restViewMode) {
            builder.humanReadable(previouslyHumanReadable);
        }
        if (version >= VERSION_START_DATE) {
            builder.timeField(Fields.START_DATE_IN_MILLIS, Fields.START_DATE, startDate);
        }
        return builder;
    }
```

```
public static License fromXContent(XContentParser parser) throws IOException {
    Builder builder = new Builder();
    XContentParser.Token token;
    while ((token = parser.nextToken()) != XContentParser.Token.END_OBJECT) {
        if (token == XContentParser.Token.FIELD_NAME) {
            String currentFieldName = parser.currentName();
            token = parser.nextToken();
            if (token.isValue()) {
                if (Fields.UID.equals(currentFieldName)) {
                    builder.uid(parser.text());
                } else if (Fields.TYPE.equals(currentFieldName)) {
                    builder.type(parser.text());
                } else if (Fields.SUBSCRIPTION_TYPE.equals(currentFieldName)) {
                    builder.subscriptionType(parser.text());
                } else if (Fields.ISSUE_DATE.equals(currentFieldName)) {
                    builder.issueDate(parseDate(parser, "issue", false));
                } else if (Fields.ISSUE_DATE_IN_MILLIS.equals(currentFieldName)) {
                    builder.issueDate(parser.longValue());
                } else if (Fields.FEATURE.equals(currentFieldName)) {
                    builder.feature(parser.text());
                } else if (Fields.EXPIRY_DATE.equals(currentFieldName)) {
                    builder.expiryDate(parseDate(parser, "expiration", true));
                } else if (Fields.EXPIRY_DATE_IN_MILLIS.equals(currentFieldName)) {
                    builder.expiryDate(parser.longValue());
                } else if (Fields.START_DATE.equals(currentFieldName)) {
                    builder.startDate(parseDate(parser, "start", false));
                } else if (Fields.START_DATE_IN_MILLIS.equals(currentFieldName)) {
                    builder.startDate(parser.longValue());
                } else if (Fields.MAX_NODES.equals(currentFieldName)) {
                    builder.maxNodes(parser.intValue());
                } else if (Fields.ISSUED_TO.equals(currentFieldName)) {
                    builder.issuedTo(parser.text());
                } else if (Fields.ISSUER.equals(currentFieldName)) {
                    builder.issuer(parser.text());
                } else if (Fields.SIGNATURE.equals(currentFieldName)) {
                    builder.signature(parser.text());
                } else if (Fields.VERSION.equals(currentFieldName)) {
                    builder.version(parser.intValue());
                }
                // Ignore unknown elements - might be new version of license
            } else if (token == XContentParser.Token.START_ARRAY) {
                // It was probably created by newer version - ignoring
                parser.skipChildren();
            } else if (token == XContentParser.Token.START_OBJECT) {
                // It was probably created by newer version - ignoring
                parser.skipChildren();
            }
            }
        }
        // not a license spec
        if (builder.signature != null) {
            byte[] signatureBytes = Base64.getDecoder().decode(builder.signature);
            ByteBuffer byteBuffer = ByteBuffer.wrap(signatureBytes);
            int version = byteBuffer.getInt();
            // we take the absolute version, because negative versions
            // mean that the license was generated by the cluster (see TrialLicense)
            // and positive version means that the license was signed
            if (version < 0) {
                version *= -1;
            }
            if (version == 0) {
                throw new ElasticsearchException("malformed signature for license [" + builder.uid + "]");
            } else if (version > VERSION_CURRENT) {
                throw new ElasticsearchException("Unknown license version found, please upgrade all nodes to the latest
" +
                        "elasticsearch-license plugin");
            }
            // signature version is the source of truth
            builder.version(version);
        }
        return builder.build();
    }

    /**
     * Returns true if the license was auto-generated (by license plugin),
     * false otherwise
     */
    public static boolean isAutoGeneratedLicense(String signature) {
        try {
            byte[] signatureBytes = Base64.getDecoder().decode(signature);
            ByteBuffer byteBuffer = ByteBuffer.wrap(signatureBytes);
            return byteBuffer.getInt() < 0;
        } catch (IllegalArgumentException e) {
            throw new IllegalStateException(e);
        }
    }

    public static License fromSource(BytesReference bytes, XContentType xContentType) throws IOException {
        if (bytes == null || bytes.length() == 0) {
            throw new ElasticsearchParseException("failed to parse license - no content provided");
        }
        if (xContentType == null) {
            throw new ElasticsearchParseException("failed to parse license - no content-type provided");
        }
        // EMPTY is safe here because we don't call namedObject
        try (InputStream byteStream = bytes.streamInput();
             XContentParser parser = xContentType.xContent()
                .createParser(NamedXContentRegistry.EMPTY, LoggingDeprecationHandler.INSTANCE, byteStream))
        {
            License license = null;
            if (parser.nextToken() == XContentParser.Token.START_OBJECT) {
                if (parser.nextToken() == XContentParser.Token.FIELD_NAME) {
                    String currentFieldName = parser.currentName();
                    if (Fields.LICENSES.equals(currentFieldName)) {
                        final List<License> pre20Licenses = new ArrayList<>();
                        if (parser.nextToken() == XContentParser.Token.START_ARRAY) {
                            while (parser.nextToken() != XContentParser.Token.END_ARRAY) {
                                pre20Licenses.add(License.fromXContent(parser));
                            }
                            // take the latest issued unexpired license
                            CollectionUtil.timSort(pre20Licenses, LATEST_ISSUE_DATE_FIRST);
                            long now = System.currentTimeMillis();
                            for (License oldLicense : pre20Licenses) {
                                if (oldLicense.expiryDate() > now) {
                                    license = oldLicense;
                                    break;
                                }
                            }
                            if (license == null && !pre20Licenses.isEmpty()) {
                                license = pre20Licenses.get(0);
                            }
                        } else {
                            throw new ElasticsearchParseException("failed to parse licenses expected an array of licenses");
                        }
                    } else if (Fields.LICENSE.equals(currentFieldName)) {
                        license = License.fromXContent(parser);
                    }
                    // Ignore all other fields - might be created with new version
                } else {
                    throw new ElasticsearchParseException("failed to parse licenses expected field");
                }
            } else {
                throw new ElasticsearchParseException("failed to parse licenses expected start object");
            }
            return license;
        }
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
```

```
        License license = (License) o;

        if (issueDate != license.issueDate) return false;
        if (expiryDate != license.expiryDate) return false;
        if (startDate!= license.startDate) return false;
        if (maxNodes != license.maxNodes) return false;
        if (version != license.version) return false;
        if (uid != null ? !uid.equals(license.uid) : license.uid != null) return false;
        if (issuer != null ? !issuer.equals(license.issuer) : license.issuer != null) return false;
        if (issuedTo != null ? !issuedTo.equals(license.issuedTo) : license.issuedTo != null) return false;
        if (type != null ? !type.equals(license.type) : license.type != null) return false;
        if (subscriptionType != null ? !subscriptionType.equals(license.subscriptionType) : license.subscriptionType !=
null)
            return false;
        if (feature != null ? !feature.equals(license.feature) : license.feature != null) return false;
        return !(signature != null ? !signature.equals(license.signature) : license.signature != null);

    }

    @Override
    public int hashCode() {
        int result = uid != null ? uid.hashCode() : 0;
        result = 31 * result + (issuer != null ? issuer.hashCode() : 0);
        result = 31 * result + (issuedTo != null ? issuedTo.hashCode() : 0);
        result = 31 * result + (int) (issueDate ^ (issueDate >>> 32));
        result = 31 * result + (type != null ? type.hashCode() : 0);
        result = 31 * result + (subscriptionType != null ? subscriptionType.hashCode() : 0);
        result = 31 * result + (feature != null ? feature.hashCode() : 0);
        result = 31 * result + (signature != null ? signature.hashCode() : 0);
        result = 31 * result + (int) (expiryDate ^ (expiryDate >>> 32));
        result = 31 * result + (int) (startDate ^ (startDate>>> 32));
        result = 31 * result + maxNodes;
        result = 31 * result + version;
        return result;
    }

    public static final class Fields {
        public static final String STATUS = "status";
        public static final String UID = "uid";
        public static final String TYPE = "type";
        public static final String SUBSCRIPTION_TYPE = "subscription_type";
        public static final String ISSUE_DATE_IN_MILLIS = "issue_date_in_millis";
        public static final String ISSUE_DATE = "issue_date";
        public static final String FEATURE = "feature";
        public static final String EXPIRY_DATE_IN_MILLIS = "expiry_date_in_millis";
        public static final String EXPIRY_DATE = "expiry_date";
        public static final String START_DATE_IN_MILLIS = "start_date_in_millis";
        public static final String START_DATE = "start_date";
        public static final String MAX_NODES = "max_nodes";
        public static final String ISSUED_TO = "issued_to";
        public static final String ISSUER = "issuer";
        public static final String VERSION = "version";
        public static final String SIGNATURE = "signature";

        public static final String LICENSES = "licenses";
        public static final String LICENSE = "license";

    }

    private static long parseDate(XContentParser parser, String description, boolean endOfTheDay) throws
IOException {
        if (parser.currentToken() == XContentParser.Token.VALUE_NUMBER) {
            return parser.longValue();
        } else {
            try {
                if (endOfTheDay) {
                    return DateUtils.endOfTheDay(parser.text());
                } else {
                    return DateUtils.beginningOfTheDay(parser.text());
                }
            } catch (IllegalArgumentException ex) {
                throw new ElasticsearchParseException("invalid " + description + " date format " + parser.text());
            }
        }
    }

    public static Builder builder() {
        return new Builder();
    }

    public static class Builder {
        private int version = License.VERSION_CURRENT;
        private String uid;
        private String issuer;
        private String issuedTo;
        private long issueDate = -1;
        private String type;
        private String subscriptionType;
        private String feature;
        private String signature;
        private long expiryDate = -1;
        private long startDate = -1;
        private int maxNodes = -1;

        public Builder uid(String uid) {
            this.uid = uid;
            return this;
        }

        public Builder version(int version) {
            this.version = version;
            return this;
        }

        public Builder issuer(String issuer) {
            this.issuer = issuer;
            return this;
        }

        public Builder issuedTo(String issuedTo) {
            this.issuedTo = issuedTo;
            return this;
        }

        public Builder issueDate(long issueDate) {
            this.issueDate = issueDate;
            return this;
        }

        public Builder type(String type) {
            this.type = type;
            return this;
        }

        public Builder subscriptionType(String subscriptionType) {
            this.subscriptionType = subscriptionType;
            return this;
        }

        public Builder feature(String feature) {
            this.feature = feature;
            return this;
        }

        public Builder expiryDate(long expiryDate) {
            this.expiryDate = expiryDate;
            return this;
        }

        public Builder maxNodes(int maxNodes) {
            this.maxNodes = maxNodes;
            return this;
        }
```

```
public Builder signature(String signature) {
    if (signature != null) {
        this.signature = signature;
    }
    return this;
}

public Builder startDate(long startDate) {
    this.startDate = startDate;
    return this;
}

public Builder fromLicenseSpec(License license, String signature) {
    return uid(license.uid())
            .version(license.version())
            .issuedTo(license.issuedTo())
            .issueDate(license.issueDate())
            .startDate(license.startDate())
            .type(license.type())
            .subscriptionType(license.subscriptionType)
            .feature(license.feature)
            .maxNodes(license.maxNodes())
            .expiryDate(license.expiryDate())
            .issuer(license.issuer())
            .signature(signature);
}

/**
 * Returns a builder that converts pre 2.0 licenses
 * to the new license format
 */
public Builder fromPre20LicenseSpec(License pre20License) {
    return uid(pre20License.uid())
            .issuedTo(pre20License.issuedTo())
            .issueDate(pre20License.issueDate())
            .maxNodes(pre20License.maxNodes())
            .expiryDate(pre20License.expiryDate());
}

public License build() {
    return new License(version, uid, issuer, issuedTo, issueDate, type,
            subscriptionType, feature, signature, expiryDate, maxNodes, startDate);
}

public Builder validate() {
    if (issuer == null) {
        throw new IllegalStateException("issuer can not be null");
    } else if (issuedTo == null) {
            throw new IllegalStateException("issuedTo can not be null");
        } else if (issueDate == -1) {
            throw new IllegalStateException("issueDate has to be set");
        } else if (type == null) {
            throw new IllegalStateException("type can not be null");
        } else if (uid == null) {
            throw new IllegalStateException("uid can not be null");
        } else if (signature == null) {
            throw new IllegalStateException("signature can not be null");
        } else if (maxNodes == -1) {
            throw new IllegalStateException("maxNodes has to be set");
        } else if (expiryDate == -1) {
            throw new IllegalStateException("expiryDate has to be set");
        }
        return this;
    }
}

public enum Status {

    ACTIVE("active"),
    INVALID("invalid"),
    EXPIRED("expired");

    private final String label;

    Status(String label) {
        this.label = label;
    }

    public String label() {
        return label;
    }

    public void writeTo(StreamOutput out) throws IOException {
        out.writeString(label);
    }

    public static Status readFrom(StreamInput in) throws IOException {
        String value = in.readString();
        switch (value) {
            case "active":
                return ACTIVE;
            case "invalid":
                return INVALID;
            case "expired":
                return EXPIRED;
            default:
                throw new IllegalArgumentException("unknown license status [" + value + "]");
            }
        }
    }

    /**
     * Returns <code>true</code> iff the license is a production licnese
     */
    public boolean isProductionLicense() {
        switch (operationMode()) {
            case MISSING:
            case TRIAL:
            case BASIC:
                return false;
            case STANDARD:
            case GOLD:
            case PLATINUM:
                return true;
            default:
                throw new AssertionError("unknown operation mode: " + operationMode());

        }
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.action.RestToXContentListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.POST;
import static org.elasticsearch.rest.RestRequest.Method.PUT;

public class RestPutLicenseAction extends XPackRestHandler {
    public RestPutLicenseAction(Settings settings, RestController controller) {
        super(settings);
        // @deprecated Remove deprecations in 6.0
        controller.registerWithDeprecatedHandler(POST, URI_BASE + "/license", this,
                                         POST, "/_license", deprecationLogger);
        controller.registerWithDeprecatedHandler(PUT, URI_BASE + "/license", this,
                                         PUT, "/_license", deprecationLogger);

        // Remove _licenses support entirely in 6.0
        controller.registerAsDeprecatedHandler(POST, "/_licenses", this,
                                       "[POST /_licenses] is deprecated! Use " +
                                       "[POST /_xpack/license] instead.",
                                       deprecationLogger);
        controller.registerAsDeprecatedHandler(PUT, "/_licenses", this,
                                       "[PUT /_licenses] is deprecated! Use " +
                                       "[PUT /_xpack/license] instead.",
                                       deprecationLogger);
    }

    @Override
    public String getName() {
        return "xpack_put_license_action";
    }

    @Override
    public RestChannelConsumer doPrepareRequest(final RestRequest request, final XPackClient client) throws
IOException {
        if (request.hasContent() == false) {
            throw new IllegalArgumentException("The license must be provided in the request body");
        }
        PutLicenseRequest putLicenseRequest = new PutLicenseRequest();
        putLicenseRequest.license(request.content(), request.getXContentType());
        putLicenseRequest.acknowledge(request.paramAsBoolean("acknowledge", false));
        putLicenseRequest.timeout(request.paramAsTime("timeout", putLicenseRequest.timeout()));
        putLicenseRequest.masterNodeTimeout(request.paramAsTime("master_timeout",
putLicenseRequest.masterNodeTimeout()));

        if ("basic".equals(putLicenseRequest.license().type())) {
            throw new IllegalArgumentException("Installing basic licenses is no longer allowed. Use the POST " +
                    "/_xpack/license/start_basic API to install a basic license that does not expire.");
        }

        return channel -> client.es().admin().cluster().execute(PutLicenseAction.INSTANCE, putLicenseRequest,
                new RestToXContentListener<>(channel));
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

public class GetLicenseResponse extends ActionResponse {

   private License license;

   GetLicenseResponse() {
   }

   GetLicenseResponse(License license) {
       this.license = license;
   }

   public License license() {
       return license;
   }

   @Override
   public void readFrom(StreamInput in) throws IOException {
       super.readFrom(in);
       if (in.readBoolean()) {
           license = License.readLicense(in);
       }
   }

   @Override
   public void writeTo(StreamOutput out) throws IOException {
       super.writeTo(out);
       if (license == null) {
           out.writeBoolean(false);
       } else {
           out.writeBoolean(true);
           license.writeTo(out);
       }
   }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.action.support.master.MasterNodeReadOperationRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

public class GetLicenseRequestBuilder extends MasterNodeReadOperationRequestBuilder<GetLicenseRequest,
GetLicenseResponse,
        GetLicenseRequestBuilder> {

    public GetLicenseRequestBuilder(ElasticsearchClient client) {
        this(client, GetLicenseAction.INSTANCE);
    }

    /**
     * Creates new get licenses request builder
     *
     * @param client elasticsearch client
     */
    public GetLicenseRequestBuilder(ElasticsearchClient client, GetLicenseAction action) {
        super(client, action, new GetLicenseRequest());
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.MasterNodeReadRequest;


public class GetLicenseRequest extends MasterNodeReadRequest<GetLicenseRequest> {

    public GetLicenseRequest() {
    }

    @Override
    public ActionRequestValidationException validate() {
        return null;
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportPostStartTrialAction extends TransportMasterNodeAction<PostStartTrialRequest,
PostStartTrialResponse> {

   private final LicenseService licenseService;

   @Inject
   public TransportPostStartTrialAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                        LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                                        IndexNameExpressionResolver indexNameExpressionResolver) {
       super(settings, PostStartTrialAction.NAME, transportService, clusterService, threadPool, actionFilters,
               indexNameExpressionResolver, PostStartTrialRequest::new);
       this.licenseService = licenseService;
   }

   @Override
   protected String executor() {
       return ThreadPool.Names.SAME;
   }

   @Override
   protected PostStartTrialResponse newResponse() {
       return new PostStartTrialResponse();
   }

   @Override
   protected void masterOperation(PostStartTrialRequest request, ClusterState state,
                                  ActionListener<PostStartTrialResponse> listener) throws Exception {
       licenseService.startTrialLicense(request, listener);
   }

   @Override
   protected ClusterBlockException checkBlock(PostStartTrialRequest request, ClusterState state) {
       return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_WRITE);
```

```
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

import static org.elasticsearch.rest.RestRequest.Method.GET;
import static org.elasticsearch.rest.RestStatus.NOT_FOUND;
import static org.elasticsearch.rest.RestStatus.OK;

public class RestGetLicenseAction extends XPackRestHandler {

  @Inject
  public RestGetLicenseAction(Settings settings, RestController controller) {
    super(settings);
    // @deprecated Remove deprecations in 6.0
    controller.registerWithDeprecatedHandler(GET,  URI_BASE + "/license", this,
                                              GET, "/_license", deprecationLogger);

    // Remove _licenses support entirely in 6.0
    controller.registerAsDeprecatedHandler(GET, "/_licenses", this,
                                            "[GET /_licenses] is deprecated! Use " +
                                            "[GET /_xpack/license] instead.",
                                            deprecationLogger);
  }

  @Override
  public String getName() {
    return "xpack_get_license_action";
    }

    /**
     * There will be only one license displayed per feature, the selected license will have the latest expiry_date
     * out of all other licenses for the feature.
     * <p>
     * The licenses are sorted by latest issue_date
     */
    @Override
    public RestChannelConsumer doPrepareRequest(final RestRequest request, final XPackClient client) throws
IOException {
        final Map<String, String> overrideParams = new HashMap<>(2);
        overrideParams.put(License.REST_VIEW_MODE, "true");
        overrideParams.put(License.LICENSE_VERSION_MODE, String.valueOf(License.VERSION_CURRENT));
        final ToXContent.Params params = new ToXContent.DelegatingMapParams(overrideParams, request);
        GetLicenseRequest getLicenseRequest = new GetLicenseRequest();
        getLicenseRequest.local(request.paramAsBoolean("local", getLicenseRequest.local()));
        return channel -> client.es().admin().cluster().execute(GetLicenseAction.INSTANCE, getLicenseRequest,
                new RestBuilderListener<GetLicenseResponse>(channel) {
                    @Override
                    public RestResponse buildResponse(GetLicenseResponse response, XContentBuilder builder) throws
Exception {
                        // Default to pretty printing, but allow ?pretty=false to disable
                        if (!request.hasParam("pretty")) {
                            builder.prettyPrint().lfAtEnd();
                        }
                        boolean hasLicense = response.license() != null;
                        builder.startObject();
                        if (hasLicense) {
                            builder.startObject("license");
                            response.license().toInnerXContent(builder, params);
                            builder.endObject();
                        }
                        builder.endObject();
                        return new BytesRestResponse(hasLicense ? OK : NOT_FOUND, builder);
                    }
                });
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.AcknowledgedRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.XContentType;

/**
* Register license request builder
*/
public class PutLicenseRequestBuilder extends AcknowledgedRequestBuilder<PutLicenseRequest,
PutLicenseResponse, PutLicenseRequestBuilder> {

  public PutLicenseRequestBuilder(ElasticsearchClient client) {
    this(client, PutLicenseAction.INSTANCE);
  }

  /**
   * Constructs register license request
   *
   * @param client elasticsearch client
   */
  public PutLicenseRequestBuilder(ElasticsearchClient client, PutLicenseAction action) {
    super(client, action, new PutLicenseRequest());
  }

  /**
   * Sets the license
   *
   * @param license license
   * @return this builder
   */
  public PutLicenseRequestBuilder setLicense(License license) {
    request.license(license);
    return this;
  }

  public PutLicenseRequestBuilder setLicense(BytesReference licenseSource, XContentType xContentType) {
    request.license(licenseSource, xContentType);
    return this;
  }

  public PutLicenseRequestBuilder setAcknowledge(boolean acknowledge) {
    request.acknowledge(acknowledge);
    return this;
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
```

```
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;
import org.elasticsearch.client.ElasticsearchClient;

public class GetBasicStatusAction extends Action<GetBasicStatusRequest, GetBasicStatusResponse,
GetBasicStatusRequestBuilder> {

  public static final GetBasicStatusAction INSTANCE = new GetBasicStatusAction();
  public static final String NAME = "cluster:admin/xpack/license/basic_status";

  private GetBasicStatusAction() {
    super(NAME);
  }

  @Override
  public GetBasicStatusRequestBuilder newRequestBuilder(ElasticsearchClient client) {
    return new GetBasicStatusRequestBuilder(client, this);
  }

  @Override
  public GetBasicStatusResponse newResponse() {
    return new GetBasicStatusResponse();
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.rest.RestStatus;

public class LicenseUtils {

  public static final String EXPIRED_FEATURE_METADATA = "es.license.expired.feature";

  /**
   * Exception to be thrown when a feature action requires a valid license, but license
   * has expired
   *
   * <code>feature</code> accessible through {@link #EXPIRED_FEATURE_METADATA} in the
   * exception's rest header
   */
  public static ElasticsearchSecurityException newComplianceException(String feature) {
    ElasticsearchSecurityException e = new ElasticsearchSecurityException("current license is non-compliant for
[{}]",
                RestStatus.FORBIDDEN, feature);
    e.addMetadata(EXPIRED_FEATURE_METADATA, feature);
    return e;
  }

  /**
   * Checks if a given {@link ElasticsearchSecurityException} refers to a feature that
   * requires a valid license, but the license has expired.
   */
  public static boolean isLicenseExpiredException(ElasticsearchSecurityException exception) {
    return (exception != null) && (exception.getMetadata(EXPIRED_FEATURE_METADATA) != null);
  }

  public static boolean licenseNeedsExtended(License license) {
    return "basic".equals(license.type()) && license.expiryDate() !=
LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequest;
import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.cluster.NamedDiff;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.common.ParseField;
import org.elasticsearch.common.io.stream.NamedWriteableRegistry;
import org.elasticsearch.common.settings.ClusterSettings;
import org.elasticsearch.common.settings.IndexScopedSettings;
import org.elasticsearch.common.settings.Setting;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.settings.SettingsFilter;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.plugins.ActionPlugin;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestHandler;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
import java.util.function.Supplier;

import static org.elasticsearch.xpack.core.XPackClientActionPlugin.isTribeNode;
import static org.elasticsearch.xpack.core.XPackPlugin.transportClientMode;

public class Licensing implements ActionPlugin {

  public static final String NAME = "license";
  protected final Settings settings;
  private final boolean isTransportClient;
  private final boolean isTribeNode;

  // Until this is moved out to its own plugin (its currently in XPackPlugin.java, we need to make sure that any edits
to this file
  // are also carried out in XPackClientPlugin.java
  public List<NamedWriteableRegistry.Entry> getNamedWriteables() {
    List<NamedWriteableRegistry.Entry> entries = new ArrayList<>();
    entries.add(new NamedWriteableRegistry.Entry(MetaData.Custom.class, LicensesMetaData.TYPE,
LicensesMetaData::new));
    entries.add(new NamedWriteableRegistry.Entry(NamedDiff.class, LicensesMetaData.TYPE,
LicensesMetaData::readDiffFrom));
    return entries;
  }

  // Until this is moved out to its own plugin (its currently in XPackPlugin.java, we need to make sure that any edits
to this file
  // are also carried out in XPackClientPlugin.java
  public List<NamedXContentRegistry.Entry> getNamedXContent() {
    List<NamedXContentRegistry.Entry> entries = new ArrayList<>();
    // Metadata
    entries.add(new NamedXContentRegistry.Entry(MetaData.Custom.class, new
ParseField(LicensesMetaData.TYPE),
        LicensesMetaData::fromXContent));
    return entries;
  }

  public Licensing(Settings settings) {
    this.settings = settings;
    isTransportClient = transportClientMode(settings);
    isTribeNode = isTribeNode(settings);
  }

  @Override
  public List<ActionHandler<? extends ActionRequest, ? extends ActionResponse>> getActions() {
    if (isTribeNode) {
      return Collections.singletonList(new ActionHandler<>(GetLicenseAction.INSTANCE,
TransportGetLicenseAction.class));
```

```
        }
        return Arrays.asList(new ActionHandler<>(PutLicenseAction.INSTANCE, TransportPutLicenseAction.class),
                new ActionHandler<>(GetLicenseAction.INSTANCE, TransportGetLicenseAction.class),
                new ActionHandler<>(DeleteLicenseAction.INSTANCE, TransportDeleteLicenseAction.class),
                new ActionHandler<>(PostStartTrialAction.INSTANCE, TransportPostStartTrialAction.class),
                new ActionHandler<>(GetTrialStatusAction.INSTANCE, TransportGetTrialStatusAction.class),
                new ActionHandler<>(PostStartBasicAction.INSTANCE, TransportPostStartBasicAction.class),
                new ActionHandler<>(GetBasicStatusAction.INSTANCE, TransportGetBasicStatusAction.class));
    }

    @Override
    public List<RestHandler> getRestHandlers(Settings settings, RestController restController, ClusterSettings
clusterSettings,
            IndexScopedSettings indexScopedSettings, SettingsFilter settingsFilter, IndexNameExpressionResolver
indexNameExpressionResolver,
            Supplier<DiscoveryNodes> nodesInCluster) {
        List<RestHandler> handlers = new ArrayList<>();
        handlers.add(new RestGetLicenseAction(settings, restController));
        if (false == isTribeNode) {
            handlers.add(new RestPutLicenseAction(settings, restController));
            handlers.add(new RestDeleteLicenseAction(settings, restController));
            handlers.add(new RestGetTrialStatus(settings, restController));
            handlers.add(new RestGetBasicStatus(settings, restController));
            handlers.add(new RestPostStartTrialLicense(settings, restController));
            handlers.add(new RestPostStartBasicLicense(settings, restController));
        }
        return handlers;
    }

    // Until this is moved out to its own plugin (its currently in XPackPlugin.java, we need to make sure that any edits
to this file
    // are also carried out in XPackClientPlugin.java
    public List<Setting<?>> getSettings() {
        // TODO convert this wildcard to a real setting
        return Collections.singletonList(Setting.groupSetting("license.", Setting.Property.NodeScope));
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;


import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.StandardCharsets;
import java.security.InvalidKeyException;
import java.security.KeyFactory;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;
import java.util.Base64;

public class CryptUtils {
  private static final int minimumPadding = 20;
  private static final byte[] salt = {
        (byte) 0xA9, (byte) 0xA2, (byte) 0xB5, (byte) 0xDE,
        (byte) 0x2A, (byte) 0x8A, (byte) 0x9A, (byte) 0xE6
  };
  private static final int iterationCount = 1024;
  private static final int aesKeyLength = 128;
  private static final String keyAlgorithm = "RSA";
  private static final String passHashAlgorithm = "SHA-512";
  private static final String DEFAULT_PASS_PHRASE = "elasticsearch-license";

  private static final SecureRandom random = new SecureRandom();

  /**
   * Read encrypted private key file content with default pass phrase
   */
  public static PrivateKey readEncryptedPrivateKey(byte[] fileContents) {
     try {
        return readEncryptedPrivateKey(fileContents, hashPassPhrase(DEFAULT_PASS_PHRASE));
     } catch (NoSuchAlgorithmException e) {
        throw new IllegalStateException(e);
     }
  }

  /**
   * Read encrypted public key file content with default pass phrase
   */
  public static PublicKey readEncryptedPublicKey(byte[] fileContents) {
    try {
        return readEncryptedPublicKey(fileContents, hashPassPhrase(DEFAULT_PASS_PHRASE));
    } catch (NoSuchAlgorithmException e) {
        throw new IllegalStateException(e);
    }
}

/**
 * Returns encrypted public key file content with default pass phrase
 */
public static byte[] writeEncryptedPublicKey(PublicKey publicKey) {
    try {
        return writeEncryptedPublicKey(publicKey, hashPassPhrase(DEFAULT_PASS_PHRASE));
    } catch (NoSuchAlgorithmException e) {
        throw new IllegalStateException(e);
    }
}

/**
 * Returns encrypted private key file content with default pass phrase
 */
public static byte[] writeEncryptedPrivateKey(PrivateKey privateKey) {
    try {
        return writeEncryptedPrivateKey(privateKey, hashPassPhrase(DEFAULT_PASS_PHRASE));
    } catch (NoSuchAlgorithmException e) {
        throw new IllegalStateException(e);
    }
}

/**
 * Read encrypted private key file content with provided <code>passPhrase</code>
 */
public static PrivateKey readEncryptedPrivateKey(byte[] fileContents, char[] passPhrase) {
    PKCS8EncodedKeySpec privateKeySpec = new PKCS8EncodedKeySpec(decrypt(fileContents, passPhrase));
    try {
        return KeyFactory.getInstance(keyAlgorithm).generatePrivate(privateKeySpec);
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        throw new IllegalStateException(e);
    }
}

/**
 * Read encrypted public key file content with provided <code>passPhrase</code>
 */
public static PublicKey readEncryptedPublicKey(byte[] fileContents, char[] passPhrase) {
    X509EncodedKeySpec publicKeySpec = new X509EncodedKeySpec(decrypt(fileContents, passPhrase));
    try {
        return KeyFactory.getInstance(CryptUtils.keyAlgorithm).generatePublic(publicKeySpec);
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        throw new IllegalStateException(e);
    }
}

/**
 * Returns encrypted public key file content with provided <code>passPhrase</code>
 */
public static byte[] writeEncryptedPublicKey(PublicKey publicKey, char[] passPhrase) {
    X509EncodedKeySpec encodedKeySpec = new X509EncodedKeySpec(publicKey.getEncoded());
    return encrypt(encodedKeySpec.getEncoded(), passPhrase);
}

/**
 * Returns encrypted private key file content with provided <code>passPhrase</code>
 */
public static byte[] writeEncryptedPrivateKey(PrivateKey privateKey, char[] passPhrase) {
    PKCS8EncodedKeySpec encodedKeySpec = new PKCS8EncodedKeySpec(privateKey.getEncoded());
    return encrypt(encodedKeySpec.getEncoded(), passPhrase);
}

/**
 * Encrypts provided <code>data</code> with <code>DEFAULT_PASS_PHRASE</code>
 */
public static byte[] encrypt(byte[] data) {
    try {
        return encrypt(data, hashPassPhrase(DEFAULT_PASS_PHRASE));
    } catch (NoSuchAlgorithmException e) {
        throw new IllegalStateException(e);
    }
}

/**
 * Decrypts provided <code>encryptedData</code> with <code>DEFAULT_PASS_PHRASE</code>
 */
public static byte[] decrypt(byte[] encryptedData) {
    try {
        return decrypt(encryptedData, hashPassPhrase(DEFAULT_PASS_PHRASE));
    } catch (NoSuchAlgorithmException e) {
        throw new IllegalStateException(e);
    }
}

/**
 * Encrypts provided <code>data</code> with <code>passPhrase</code>
 */
public static byte[] encrypt(byte[] data, char[] passPhrase) {
    try {
        final Cipher encryptionCipher = getEncryptionCipher(getSecretKey(passPhrase));
        return encryptionCipher.doFinal(pad(data, minimumPadding));
    } catch (InvalidKeySpecException | IllegalBlockSizeException | BadPaddingException e) {
        throw new IllegalStateException(e);
    }
}

/**
 * Decrypts provided <code>encryptedData</code> with <code>passPhrase</code>
 */
private static byte[] decrypt(byte[] encryptedData, char[] passPhrase) {
    try {
        final Cipher cipher = getDecryptionCipher(getSecretKey(passPhrase));
        return unPad(cipher.doFinal(encryptedData));
    } catch (IllegalBlockSizeException | BadPaddingException | InvalidKeySpecException e) {
        throw new IllegalStateException(e);
    }

}

private static SecretKey getSecretKey(char[] passPhrase) throws InvalidKeySpecException {
    try {
        PBEKeySpec keySpec = new PBEKeySpec(passPhrase, salt, iterationCount, aesKeyLength);

        byte[] shortKey = SecretKeyFactory.getInstance("PBEWithSHA1AndDESede").
                generateSecret(keySpec).getEncoded();

        byte[] intermediaryKey = new byte[aesKeyLength / 8];
        for (int i = 0, j = 0; i < aesKeyLength / 8; i++) {
            intermediaryKey[i] = shortKey[j];
            if (++j == shortKey.length)
                j = 0;
        }

        return new SecretKeySpec(intermediaryKey, "AES");
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        throw new IllegalStateException(e);
    }
}

private static Cipher getEncryptionCipher(SecretKey secretKey) {
    return getCipher(Cipher.ENCRYPT_MODE, secretKey);
}

private static Cipher getDecryptionCipher(SecretKey secretKey) {
    return getCipher(Cipher.DECRYPT_MODE, secretKey);
}
```

```
private static Cipher getCipher(int mode, SecretKey secretKey) {
    try {
        Cipher cipher = Cipher.getInstance(secretKey.getAlgorithm());
        cipher.init(mode, secretKey, random);
        return cipher;
    } catch (NoSuchAlgorithmException | InvalidKeyException | NoSuchPaddingException e) {
        throw new IllegalStateException(e);
    }
}

private static byte[] pad(byte[] bytes, int length) {
    if (bytes.length >= length) {
        byte[] out = new byte[bytes.length + 1];
        System.arraycopy(bytes, 0, out, 0, bytes.length);
        out[bytes.length] = (byte) 1;
        return out;
    }

    byte[] out = new byte[length + 1];

    int i = 0;
    for (; i < bytes.length; i++)
        out[i] = bytes[i];

    int padded = length - i;

    // fill the rest with random bytes
    byte[] fill = new byte[padded - 1];
    random.nextBytes(fill);
    System.arraycopy(fill, 0, out, i, padded - 1);

    out[length] = (byte) (padded + 1);

    return out;
}

private static byte[] unPad(byte[] bytes) {
    int padded = (int) bytes[bytes.length - 1];
    int targetLength = bytes.length - padded;

    byte[] out = new byte[targetLength];

    System.arraycopy(bytes, 0, out, 0, targetLength);

    return out;
}

private static char[] hashPassPhrase(String passPhrase) throws NoSuchAlgorithmException {
        final byte[] passBytes = passPhrase.getBytes(StandardCharsets.UTF_8);
        final byte[] digest = MessageDigest.getInstance(passHashAlgorithm).digest(passBytes);
        return Base64.getEncoder().encodeToString(digest).toCharArray();
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.rest.RestStatus;

import java.io.IOException;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

class PostStartTrialResponse extends ActionResponse {

    // Nodes Prior to 6.3 did not have NEED_ACKNOWLEDGEMENT as part of status
    enum Pre63Status {
        UPGRADED_TO_TRIAL,
        TRIAL_ALREADY_ACTIVATED;
    }
    enum Status {
        UPGRADED_TO_TRIAL(true, null, RestStatus.OK),
        TRIAL_ALREADY_ACTIVATED(false, "Operation failed: Trial was already activated.",
RestStatus.FORBIDDEN),
        NEED_ACKNOWLEDGEMENT(false,"Operation failed: Needs acknowledgement.", RestStatus.OK);

        private final boolean isTrialStarted;

        private final String errorMessage;
        private final RestStatus restStatus;
        Status(boolean isTrialStarted, String errorMessage, RestStatus restStatus) {
            this.isTrialStarted = isTrialStarted;
            this.errorMessage = errorMessage;
            this.restStatus = restStatus;
        }

        boolean isTrialStarted() {
            return isTrialStarted;
        }

        String getErrorMessage() {
            return errorMessage;
        }

        RestStatus getRestStatus() {
            return restStatus;
        }

    }

    private Status status;
    private Map<String, String[]> acknowledgeMessages;
    private String acknowledgeMessage;

    PostStartTrialResponse() {
    }

    PostStartTrialResponse(Status status) {
        this(status, Collections.emptyMap(), null);
    }

    PostStartTrialResponse(Status status, Map<String, String[]> acknowledgeMessages, String acknowledgeMessage)
{
        this.status = status;
        this.acknowledgeMessages = acknowledgeMessages;
        this.acknowledgeMessage = acknowledgeMessage;
    }

    public Status getStatus() {
        return status;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        status = in.readEnum(Status.class);
        if (in.getVersion().onOrAfter(Version.V_6_3_0)) {
            acknowledgeMessage = in.readOptionalString();
            int size = in.readVInt();
            Map<String, String[]> acknowledgeMessages = new HashMap<>(size);
            for (int i = 0; i < size; i++) {
                String feature = in.readString();
                int nMessages = in.readVInt();
                String[] messages = new String[nMessages];
                for (int j = 0; j < nMessages; j++) {
                    messages[j] = in.readString();
                }
                acknowledgeMessages.put(feature, messages);
            }
            this.acknowledgeMessages = acknowledgeMessages;
        } else {
            this.acknowledgeMessages = Collections.emptyMap();
        }
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        Version version = Version.V_6_3_0;
        if (out.getVersion().onOrAfter(version)) {
            out.writeEnum(status);
            out.writeOptionalString(acknowledgeMessage);
            out.writeVInt(acknowledgeMessages.size());
            for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
                out.writeString(entry.getKey());
                out.writeVInt(entry.getValue().length);
                for (String message : entry.getValue()) {
                    out.writeString(message);
                }
            }
        } else {
            if (status == Status.UPGRADED_TO_TRIAL) {
                out.writeEnum(Pre63Status.UPGRADED_TO_TRIAL);
            } else if (status == Status.TRIAL_ALREADY_ACTIVATED) {
                out.writeEnum(Pre63Status.TRIAL_ALREADY_ACTIVATED);
            } else {
                throw new IllegalArgumentException("Starting trial on node with version [" + Version.CURRENT + "]
requires " +
                        "acknowledgement parameter.");
            }
        }
    }

    Map<String, String[]> getAcknowledgementMessages() {
        return acknowledgeMessages;
    }

    String getAcknowledgementMessage() {
        return acknowledgeMessage;
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
```

```
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.io.stream.Writeable;
import org.elasticsearch.common.xcontent.ToXContentObject;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.xpack.core.XPackBuild;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.Set;
import java.util.stream.Collectors;

public class XPackInfoResponse extends ActionResponse {

    @Nullable private BuildInfo buildInfo;
    @Nullable private LicenseInfo licenseInfo;
    @Nullable private FeatureSetsInfo featureSetsInfo;

    public XPackInfoResponse() {}

    public XPackInfoResponse(@Nullable BuildInfo buildInfo, @Nullable LicenseInfo licenseInfo, @Nullable
FeatureSetsInfo featureSetsInfo) {
        this.buildInfo = buildInfo;
        this.licenseInfo = licenseInfo;
        this.featureSetsInfo = featureSetsInfo;
    }

    /**
     * @return  The build info (incl. build hash and timestamp)
     */
    public BuildInfo getBuildInfo() {
        return buildInfo;
    }

    /**
     * @return  The current license info (incl. UID, type/mode. status and expiry date). May return {@code null}
when no
     *          license is currently installed.
     */
    public LicenseInfo getLicenseInfo() {
        return licenseInfo;
    }

    /**
     * @return  The current status of the feature sets in X-Pack. Feature sets describe the features available/enabled in
X-Pack.
     */
    public FeatureSetsInfo getFeatureSetsInfo() {
        return featureSetsInfo;
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        super.writeTo(out);
        out.writeOptionalWriteable(buildInfo);
        out.writeOptionalWriteable(licenseInfo);
        out.writeOptionalWriteable(featureSetsInfo);
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        this.buildInfo = in.readOptionalWriteable(BuildInfo::new);
        this.licenseInfo = in.readOptionalWriteable(LicenseInfo::new);
        this.featureSetsInfo = in.readOptionalWriteable(FeatureSetsInfo::new);
    }

    public static class LicenseInfo implements ToXContentObject, Writeable {

        private final String uid;
        private final String type;
        private final String mode;
        private final long expiryDate;
        private final License.Status status;

        public LicenseInfo(License license) {
            this(license.uid(), license.type(), license.operationMode().name().toLowerCase(Locale.ROOT),
                    license.status(), license.expiryDate());
        }

        public LicenseInfo(StreamInput in) throws IOException {
            this(in.readString(), in.readString(), in.readString(), License.Status.readFrom(in), in.readLong());
        }

        public LicenseInfo(String uid, String type, String mode, License.Status status, long expiryDate) {
            this.uid = uid;
            this.type = type;
            this.mode = mode;
            this.status = status;
            this.expiryDate = expiryDate;
        }

        public String getUid() {
            return uid;
        }

        public String getType() {
            return type;
        }

        public String getMode() {
            return mode;
        }

        public long getExpiryDate() {
            return expiryDate;
        }

        public License.Status getStatus() {
            return status;
        }

        @Override
        public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
            builder.startObject()
                .field("uid", uid)
                .field("type", type)
                .field("mode", mode)
                .field("status", status.label());
            if (expiryDate != LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS) {
                builder.timeField("expiry_date_in_millis", "expiry_date", expiryDate);
            }
            return builder.endObject();
        }

        public void writeTo(StreamOutput out) throws IOException {
            out.writeString(uid);
            out.writeString(type);
            out.writeString(mode);
            status.writeTo(out);
            out.writeLong(expiryDate);
        }
    }

    public static class BuildInfo implements ToXContentObject, Writeable {
```

```
    private final String hash;
    private final String timestamp;

    public BuildInfo(XPackBuild build) {
        this(build.shortHash(), build.date());
    }

    public BuildInfo(StreamInput input) throws IOException {
        this(input.readString(), input.readString());
    }

    public BuildInfo(String hash, String timestamp) {
        this.hash = hash;
        this.timestamp = timestamp;
    }

    public String getHash() {
        return hash;
    }

    public String getTimestamp() {
        return timestamp;
    }

    @Override
    public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
        return builder.startObject()
                .field("hash", hash)
                .field("date", timestamp)
                .endObject();
    }

    public void writeTo(StreamOutput output) throws IOException {
        output.writeString(hash);
        output.writeString(timestamp);
    }
}

public static class FeatureSetsInfo implements ToXContentObject, Writeable {

    private final Map<String, FeatureSet> featureSets;

    public FeatureSetsInfo(StreamInput in) throws IOException {
        int size = in.readVInt();
        Map<String, FeatureSet> featureSets = new HashMap<>(size);
        for (int i = 0; i < size; i++) {
            FeatureSet featureSet = new FeatureSet(in);
            featureSets.put(featureSet.name, featureSet);
        }
        this.featureSets = Collections.unmodifiableMap(featureSets);
    }

    public FeatureSetsInfo(Set<FeatureSet> featureSets) {
        Map<String, FeatureSet> map = new HashMap<>(featureSets.size());
        for (FeatureSet featureSet : featureSets) {
            map.put(featureSet.name, featureSet);
        }
        this.featureSets = Collections.unmodifiableMap(map);
    }

    public Map<String, FeatureSet> getFeatureSets() {
        return featureSets;
    }

    @Override
    public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
        builder.startObject();
        List<String> names = new
ArrayList<>(this.featureSets.keySet()).stream().sorted().collect(Collectors.toList());
        for (String name : names) {
            builder.field(name, featureSets.get(name), params);
        }
        return builder.endObject();
    }

    public void writeTo(StreamOutput out) throws IOException {
        out.writeVInt(featureSets.size());
        for (FeatureSet featureSet : featureSets.values()) {
            featureSet.writeTo(out);
        }
    }

    public static class FeatureSet implements ToXContentObject, Writeable {

        private final String name;
        @Nullable private final String description;
        private final boolean available;
        private final boolean enabled;
        @Nullable private final Map<String, Object> nativeCodeInfo;

        public FeatureSet(StreamInput in) throws IOException {
            this(in.readString(), in.readOptionalString(), in.readBoolean(), in.readBoolean(),
                    in.getVersion().onOrAfter(Version.V_5_4_0) ? in.readMap() : null);
        }
```

```
public FeatureSet(String name, @Nullable String description, boolean available, boolean enabled,
         @Nullable Map<String, Object> nativeCodeInfo) {
    this.name = name;
    this.description = description;
    this.available = available;
    this.enabled = enabled;
    this.nativeCodeInfo = nativeCodeInfo;
}

public String name() {
    return name;
}

@Nullable
public String description() {
    return description;
}

public boolean available() {
    return available;
}

public boolean enabled() {
    return enabled;
}

@Nullable
public Map<String, Object> nativeCodeInfo() {
    return nativeCodeInfo;
}

public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
    builder.startObject();
    if (description != null) {
        builder.field("description", description);
    }
    builder.field("available", available);
    builder.field("enabled", enabled);
    if (nativeCodeInfo != null) {
        builder.field("native_code_info", nativeCodeInfo);
    }
    return builder.endObject();
}

public void writeTo(StreamOutput out) throws IOException {
    out.writeString(name);
    out.writeOptionalString(description);
    out.writeBoolean(available);
```

```
            out.writeBoolean(enabled);
            if (out.getVersion().onOrAfter(Version.V_5_4_0)) {
                out.writeMap(nativeCodeInfo);
            }
        }
    }

  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;
import org.elasticsearch.client.ElasticsearchClient;

public class PostStartBasicAction extends Action<PostStartBasicRequest, PostStartBasicResponse,
PostStartBasicRequestBuilder> {

  public static final PostStartBasicAction INSTANCE = new PostStartBasicAction();
  public static final String NAME = "cluster:admin/xpack/license/start_basic";

  private PostStartBasicAction() {
      super(NAME);
  }

  @Override
  public PostStartBasicRequestBuilder newRequestBuilder(ElasticsearchClient client) {
      return new PostStartBasicRequestBuilder(client, this);
  }

  @Override
  public PostStartBasicResponse newResponse() {
      return new PostStartBasicResponse();
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;
```

```
class GetBasicStatusRequestBuilder extends ActionRequestBuilder<GetBasicStatusRequest,
GetBasicStatusResponse,
        GetBasicStatusRequestBuilder> {

    GetBasicStatusRequestBuilder(ElasticsearchClient client, GetBasicStatusAction action) {
        super(client, action, new GetBasicStatusRequest());
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.MasterNodeRequest;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

public class PostStartTrialRequest extends MasterNodeRequest<PostStartTrialRequest> {

    private boolean acknowledge = false;
    private String type;

    @Override
    public ActionRequestValidationException validate() {
        return null;
    }

    public PostStartTrialRequest setType(String type) {
        this.type = type;
        return this;
    }

    public String getType() {
        return type;
    }

    public PostStartTrialRequest acknowledge(boolean acknowledge) {
        this.acknowledge = acknowledge;
        return this;
    }
    public boolean isAcknowledged() {
        return acknowledge;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        super.readFrom(in);
        if (in.getVersion().onOrAfter(Version.V_6_3_0)) {
            type = in.readString();
            acknowledge = in.readBoolean();
        } else {
            type = "trial";
            acknowledge = true;
        }
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        Version version = Version.V_6_3_0;
        if (out.getVersion().onOrAfter(version)) {
            super.writeTo(out);
            out.writeString(type);
            out.writeBoolean(acknowledge);
        } else {
            if ("trial".equals(type) == false) {
                throw new IllegalArgumentException("All nodes in cluster must be version [" + version
                        + "] or newer to start trial with a different type than 'trial'. Attempting to write to " +
                        "a node with version [" + out.getVersion() + "] with trial type [" + type + "].");
            } else if (acknowledge == false) {
                throw new IllegalArgumentException("Request must be acknowledged to send to a node with a version " +
                        "prior to [" + version + "]. Attempting to send request to node with version [" + out.getVersion() + "] "
+
                        "without acknowledgement.");
            } else {
                super.writeTo(out);
            }
        }
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeReadAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportGetTrialStatusAction extends TransportMasterNodeReadAction<GetTrialStatusRequest,
GetTrialStatusResponse> {

  @Inject
  public TransportGetTrialStatusAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                       ThreadPool threadPool, ActionFilters actionFilters,
                                       IndexNameExpressionResolver indexNameExpressionResolver) {
    super(settings, GetTrialStatusAction.NAME, transportService, clusterService, threadPool, actionFilters,
        indexNameExpressionResolver, GetTrialStatusRequest::new);
  }

  @Override
  protected String executor() {
    return ThreadPool.Names.SAME;
  }

  @Override
  protected GetTrialStatusResponse newResponse() {
    return new GetTrialStatusResponse();
  }

  @Override
  protected void masterOperation(GetTrialStatusRequest request, ClusterState state,
                                 ActionListener<GetTrialStatusResponse> listener) throws Exception {
    LicensesMetaData licensesMetaData = state.metaData().custom(LicensesMetaData.TYPE);
    listener.onResponse(new GetTrialStatusResponse(licensesMetaData == null ||
licensesMetaData.isEligibleForTrial()));

  }

  @Override
  protected ClusterBlockException checkBlock(GetTrialStatusRequest request, ClusterState state) {
    return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_READ);
  }
}
```

```
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.Version;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.AckedClusterStateUpdateTask;
import org.elasticsearch.cluster.ClusterChangedEvent;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateListener;
import org.elasticsearch.cluster.ack.ClusterStateUpdateResponse;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.component.AbstractLifecycleComponent;
import org.elasticsearch.common.component.Lifecycle;
import org.elasticsearch.common.joda.FormatDateTimeFormatter;
import org.elasticsearch.common.joda.Joda;
import org.elasticsearch.common.logging.LoggerMessageFormat;
import org.elasticsearch.common.settings.Setting;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.discovery.DiscoveryModule;
import org.elasticsearch.env.Environment;
import org.elasticsearch.gateway.GatewayService;
import org.elasticsearch.watcher.ResourceWatcherService;
import org.elasticsearch.xpack.core.XPackPlugin;
import org.elasticsearch.xpack.core.XPackSettings;
import org.elasticsearch.xpack.core.scheduler.SchedulerEngine;

import java.time.Clock;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.Set;
import java.util.concurrent.atomic.AtomicReference;

/**
* Service responsible for managing {@link LicensesMetaData}.
* <p>
* On the master node, the service handles updating the cluster state when a new license is registered.
* It also listens on all nodes for cluster state updates, and updates {@link XPackLicenseState} when
* the license changes are detected in the cluster state.
*/
public class LicenseService extends AbstractLifecycleComponent implements ClusterStateListener,
SchedulerEngine.Listener {

  public static final Setting<String> SELF_GENERATED_LICENSE_TYPE = new
Setting<>("xpack.license.self_generated.type",
        (s) -> "basic", (s) -> {
      if (SelfGeneratedLicense.validSelfGeneratedType(s)) {
        return s;
      } else {
        throw new IllegalArgumentException("Illegal self generated license type [" + s + "]. Must be trial or basic.");
      }
  }, Setting.Property.NodeScope);

  // pkg private for tests
  static final TimeValue NON_BASIC_SELF_GENERATED_LICENSE_DURATION =
TimeValue.timeValueHours(30 * 24);

  static final Set<String> VALID_TRIAL_TYPES = new HashSet<>(Arrays.asList("trial", "platinum", "gold"));

  /**
   * Duration of grace period after a license has expired
   */
  static final TimeValue GRACE_PERIOD_DURATION = days(7);

  public static final long BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS =
Long.MAX_VALUE - days(365).millis();

  private final ClusterService clusterService;

  /**
   * The xpack feature state to update when license changes are made.
   */
  private final XPackLicenseState licenseState;

  /**
   * Currently active license
   */
  private final AtomicReference<License> currentLicense = new AtomicReference<>();
  private SchedulerEngine scheduler;
  private final Clock clock;

  /**
   * File watcher for operation mode changes
```

```
     */
    private final OperationModeFileWatcher operationModeFileWatcher;

    /**
     * Callbacks to notify relative to license expiry
     */
    private List<ExpirationCallback> expirationCallbacks = new ArrayList<>();

    /**
     * Max number of nodes licensed by generated trial license
     */
    static final int SELF_GENERATED_LICENSE_MAX_NODES = 1000;

    public static final String LICENSE_JOB = "licenseJob";

    private static final FormatDateTimeFormatter DATE_FORMATTER = Joda.forPattern("EEEE, MMMMM dd,
yyyy", Locale.ROOT);

    private static final String ACKNOWLEDGEMENT_HEADER = "This license update requires acknowledgement.
To acknowledge the license, " +
            "please read the following messages and update the license again, this time with the \"acknowledge=true\"
parameter:";

    public LicenseService(Settings settings, ClusterService clusterService, Clock clock, Environment env,
                          ResourceWatcherService resourceWatcherService, XPackLicenseState licenseState) {
        super(settings);
        this.clusterService = clusterService;
        this.clock = clock;
        this.scheduler = new SchedulerEngine(clock);
        this.licenseState = licenseState;
        this.operationModeFileWatcher = new OperationModeFileWatcher(resourceWatcherService,
                XPackPlugin.resolveConfigFile(env, "license_mode"), logger, () -> updateLicenseState(getLicense()));
        this.scheduler.register(this);
        populateExpirationCallbacks();
    }

    private void logExpirationWarning(long expirationMillis, boolean expired) {
        String expiredMsg = expired ? "expired" : "will expire";
        String general = LoggerMessageFormat.format(null, "License [{}] on [{}].\n" +
                "# If you have a new license, please update it. Otherwise, please reach out to\n" +
                "# your support contact.\n" +
                "# ", expiredMsg, DATE_FORMATTER.printer().print(expirationMillis));
        if (expired) {
            general = general.toUpperCase(Locale.ROOT);
        }
        StringBuilder builder = new StringBuilder(general);
        builder.append(System.lineSeparator());
        if (expired) {
            builder.append("# COMMERCIAL PLUGINS OPERATING WITH REDUCED FUNCTIONALITY");
        } else {
            builder.append("# Commercial plugins operate with reduced functionality on license expiration:");
        }
        XPackLicenseState.EXPIRATION_MESSAGES.forEach((feature, messages) -> {
            if (messages.length > 0) {
                builder.append(System.lineSeparator());
                builder.append("# - ");
                builder.append(feature);
                for (String message : messages) {
                    builder.append(System.lineSeparator());
                    builder.append("#  - ");
                    builder.append(message);
                }
            }
        });
        logger.warn("{}", builder);
    }

    private void populateExpirationCallbacks() {
        expirationCallbacks.add(new ExpirationCallback.Pre(days(7), days(25), days(1)) {
            @Override
            public void on(License license) {
                logExpirationWarning(license.expiryDate(), false);
            }
        });
        expirationCallbacks.add(new ExpirationCallback.Pre(days(0), days(7), TimeValue.timeValueMinutes(10)) {
            @Override
            public void on(License license) {
                logExpirationWarning(license.expiryDate(), false);
            }
        });
        expirationCallbacks.add(new ExpirationCallback.Post(days(0), null, TimeValue.timeValueMinutes(10)) {
            @Override
            public void on(License license) {
                // logged when grace period begins
                logExpirationWarning(license.expiryDate(), true);
            }
        });
    }

    /**
     * Registers new license in the cluster
     * Master only operation. Installs a new license on the master provided it is VALID
     */
    public void registerLicense(final PutLicenseRequest request, final ActionListener<PutLicenseResponse> listener)
{
        final License newLicense = request.license();
        final long now = clock.millis();
        if (!LicenseVerifier.verifyLicense(newLicense) || newLicense.issueDate() > now || newLicense.startDate() >
now) {
            listener.onResponse(new PutLicenseResponse(true, LicensesStatus.INVALID));
        } else if (newLicense.type().equals("basic")) {
            listener.onFailure(new IllegalArgumentException("Registering basic licenses is not allowed."));
        } else if (newLicense.expiryDate() < now) {
            listener.onResponse(new PutLicenseResponse(true, LicensesStatus.EXPIRED));
        } else {
            if (!request.acknowledged()) {
                // TODO: ack messages should be generated on the master, since another node's cluster state may be
behind...
                final License currentLicense = getLicense();
                if (currentLicense != null) {
                    Map<String, String[]> acknowledgeMessages = getAckMessages(newLicense, currentLicense);
                    if (acknowledgeMessages.isEmpty() == false) {
                        // needs acknowledgement
                        listener.onResponse(new PutLicenseResponse(false, LicensesStatus.VALID,
ACKNOWLEDGEMENT_HEADER,
                                acknowledgeMessages));
                        return;
                    }
                }
            }

            if (newLicense.isProductionLicense()
                    && XPackSettings.SECURITY_ENABLED.get(settings)
                    && XPackSettings.TRANSPORT_SSL_ENABLED.get(settings) == false
                    && isProductionMode(settings, clusterService.localNode())) {
                // security is on but TLS is not configured we gonna fail the entire request and throw an exception
                throw new IllegalStateException("Cannot install a [" + newLicense.operationMode() +
                        "] license unless TLS is configured or security is disabled");
                // TODO we should really validate that all nodes have xpack installed and are consistently configured but
this
                // should happen on a different level and not in this code
            } else {
                clusterService.submitStateUpdateTask("register license [" + newLicense.uid() + "]", new
                        AckedClusterStateUpdateTask<PutLicenseResponse>(request, listener) {
                            @Override
                            protected PutLicenseResponse newResponse(boolean acknowledged) {
                                return new PutLicenseResponse(acknowledged, LicensesStatus.VALID);
                            }

                            @Override
                            public ClusterState execute(ClusterState currentState) throws Exception {
                                XPackPlugin.checkReadyForXPackCustomMetadata(currentState);
                                MetaData currentMetadata = currentState.metaData();
                                LicensesMetaData licensesMetaData = currentMetadata.custom(LicensesMetaData.TYPE);
                                Version trialVersion = null;
                                if (licensesMetaData != null) {
                                    trialVersion = licensesMetaData.getMostRecentTrialVersion();
                                }
                                MetaData.Builder mdBuilder = MetaData.builder(currentMetadata);
                                mdBuilder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(newLicense,
trialVersion));
                                return ClusterState.builder(currentState).metaData(mdBuilder).build();
                            }
                        });
            }
        }
    }

    public static Map<String, String[]> getAckMessages(License newLicense, License currentLicense) {
        Map<String, String[]> acknowledgeMessages = new HashMap<>();
        if (!License.isAutoGeneratedLicense(currentLicense.signature()) // current license is not auto-generated
                && currentLicense.issueDate() > newLicense.issueDate()) { // and has a later issue date
            acknowledgeMessages.put("license", new String[]{
                    "The new license is older than the currently installed license. " +
                            "Are you sure you want to override the current license?"});
        }
        XPackLicenseState.ACKNOWLEDGMENT_MESSAGES.forEach((feature, ackMessages) -> {
            String[] messages = ackMessages.apply(currentLicense.operationMode(), newLicense.operationMode());
            if (messages.length > 0) {
                acknowledgeMessages.put(feature, messages);
            }
        });
        return acknowledgeMessages;
    }


    private static TimeValue days(int days) {
        return TimeValue.timeValueHours(days * 24);
    }

    @Override
    public void triggered(SchedulerEngine.Event event) {
        final LicensesMetaData licensesMetaData =
clusterService.state().metaData().custom(LicensesMetaData.TYPE);
        if (licensesMetaData != null) {
            final License license = licensesMetaData.getLicense();
            if (event.getJobName().equals(LICENSE_JOB)) {
                updateLicenseState(license);
            } else if (event.getJobName().startsWith(ExpirationCallback.EXPIRATION_JOB_PREFIX)) {
                expirationCallbacks.stream()
                        .filter(expirationCallback -> expirationCallback.getId().equals(event.getJobName()))
                        .forEach(expirationCallback -> expirationCallback.on(license));
            }
        }
    }

    /**
     * Remove license from the cluster state metadata
     */
    public void removeLicense(final DeleteLicenseRequest request, final
ActionListener<ClusterStateUpdateResponse> listener) {
        clusterService.submitStateUpdateTask("delete license",
                new AckedClusterStateUpdateTask<ClusterStateUpdateResponse>(request, listener) {
                    @Override
                    protected ClusterStateUpdateResponse newResponse(boolean acknowledged) {
                        return new ClusterStateUpdateResponse(acknowledged);
                    }

                    @Override
                    public ClusterState execute(ClusterState currentState) throws Exception {
                        MetaData metaData = currentState.metaData();
                        final LicensesMetaData currentLicenses = metaData.custom(LicensesMetaData.TYPE);
                        if (currentLicenses.getLicense() != LicensesMetaData.LICENSE_TOMBSTONE) {
                            MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
                            LicensesMetaData newMetadata = new
LicensesMetaData(LicensesMetaData.LICENSE_TOMBSTONE,
                                    currentLicenses.getMostRecentTrialVersion());
                            mdBuilder.putCustom(LicensesMetaData.TYPE, newMetadata);
                            return ClusterState.builder(currentState).metaData(mdBuilder).build();
                        } else {
                            return currentState;
                        }
                    }
                });
    }

    public License getLicense() {
        final License license = getLicense(clusterService.state().metaData());
        return license == LicensesMetaData.LICENSE_TOMBSTONE ? null : license;
    }

    void startTrialLicense(PostStartTrialRequest request, final ActionListener<PostStartTrialResponse> listener) {
        if (VALID_TRIAL_TYPES.contains(request.getType()) == false) {
            throw new IllegalArgumentException("Cannot start trial of type [" + request.getType() + "]. Valid trial types
are "
                    + VALID_TRIAL_TYPES + ".");
        }
        StartTrialClusterTask task = new StartTrialClusterTask(logger, clusterService.getClusterName().value(), clock,
request, listener);
        clusterService.submitStateUpdateTask("started trial license", task);
    }

    void startBasicLicense(PostStartBasicRequest request, final ActionListener<PostStartBasicResponse> listener) {
        StartBasicClusterTask task = new StartBasicClusterTask(logger, clusterService.getClusterName().value(),
clock, request, listener);
        clusterService.submitStateUpdateTask("start basic license", task);
    }

    /**
     * Master-only operation to generate a one-time global self generated license.
     * The self generated license is only generated and stored if the current cluster state metadata
     * has no existing license. If the cluster currently has a basic license that has an expiration date,
     * a new basic license with no expiration date is generated.
     */
    private void registerOrUpdateSelfGeneratedLicense() {
        clusterService.submitStateUpdateTask("maybe generate license for cluster",
                new StartupSelfGeneratedLicenseTask(settings, clock, clusterService));
    }

    @Override
    protected void doStart() throws ElasticsearchException {
        clusterService.addListener(this);
        scheduler.start(Collections.emptyList());
        logger.debug("initializing license state");
        if (clusterService.lifecycleState() == Lifecycle.State.STARTED) {
            final ClusterState clusterState = clusterService.state();
            if (clusterState.blocks().hasGlobalBlock(GatewayService.STATE_NOT_RECOVERED_BLOCK) == false
&&
                    clusterState.nodes().getMasterNode() != null &&
XPackPlugin.isReadyForXPackCustomMetadata(clusterState)) {
                final LicensesMetaData currentMetaData = clusterState.metaData().custom(LicensesMetaData.TYPE);
                boolean noLicense = currentMetaData == null || currentMetaData.getLicense() == null;
                if (clusterState.getNodes().isLocalNodeElectedMaster() &&
                        (noLicense || LicenseUtils.licenseNeedsExtended(currentMetaData.getLicense()))) {
                    // triggers a cluster changed event eventually notifying the current licensee
                    registerOrUpdateSelfGeneratedLicense();
                }
            }
        }
    }

    @Override
    protected void doStop() throws ElasticsearchException {
        clusterService.removeListener(this);
        scheduler.stop();
        // clear current license
        currentLicense.set(null);
    }
```

```
  @Override
  protected void doClose() throws ElasticsearchException {
  }

  /**
   * When there is no global block on {@link
org.elasticsearch.gateway.GatewayService#STATE_NOT_RECOVERED_BLOCK}
   * notify licensees and issue auto-generated license if no license has been installed/issued yet.
   */
  @Override
  public void clusterChanged(ClusterChangedEvent event) {
    final ClusterState previousClusterState = event.previousState();
    final ClusterState currentClusterState = event.state();
    if (!currentClusterState.blocks().hasGlobalBlock(GatewayService.STATE_NOT_RECOVERED_BLOCK)) {
      if (XPackPlugin.isReadyForXPackCustomMetadata(currentClusterState) == false) {
        logger.debug("cannot add license to cluster as the following nodes might not understand the license
metadata: {}",
          () -> XPackPlugin.nodesNotReadyForXPackCustomMetadata(currentClusterState));
        return;
      }

      final LicensesMetaData prevLicensesMetaData =
previousClusterState.getMetaData().custom(LicensesMetaData.TYPE);
      final LicensesMetaData currentLicensesMetaData =
currentClusterState.getMetaData().custom(LicensesMetaData.TYPE);
      if (logger.isDebugEnabled()) {
        logger.debug("previous [{}]", prevLicensesMetaData);
        logger.debug("current [{}]", currentLicensesMetaData);
      }
      // notify all interested plugins
      if (previousClusterState.blocks().hasGlobalBlock(GatewayService.STATE_NOT_RECOVERED_BLOCK)
          || prevLicensesMetaData == null) {
        if (currentLicensesMetaData != null) {
          onUpdate(currentLicensesMetaData);
        }
      } else if (!prevLicensesMetaData.equals(currentLicensesMetaData)) {
        onUpdate(currentLicensesMetaData);
      }

      License currentLicense = null;
      boolean noLicenseInPrevMetadata = prevLicensesMetaData == null || prevLicensesMetaData.getLicense()
== null;
      if (noLicenseInPrevMetadata == false) {
        currentLicense = prevLicensesMetaData.getLicense();
      }
      boolean noLicenseInCurrentMetadata = (currentLicensesMetaData == null ||
currentLicensesMetaData.getLicense() == null);
```

```
            if (noLicenseInCurrentMetadata == false) {
                currentLicense = currentLicensesMetaData.getLicense();
            }

            boolean noLicense = noLicenseInPrevMetadata && noLicenseInCurrentMetadata;
            // auto-generate license if no licenses ever existed or if the current license is basic and
            // needs extended. this will trigger a subsequent cluster changed event
            if (currentClusterState.getNodes().isLocalNodeElectedMaster()
                    && (noLicense || LicenseUtils.licenseNeedsExtended(currentLicense))) {
                registerOrUpdateSelfGeneratedLicense();
            }
        } else if (logger.isDebugEnabled()) {
            logger.debug("skipped license notifications reason: [{}]",
GatewayService.STATE_NOT_RECOVERED_BLOCK);
        }
    }

    protected void updateLicenseState(final License license) {
        if (license == LicensesMetaData.LICENSE_TOMBSTONE) {
            // implies license has been explicitly deleted
            licenseState.update(License.OperationMode.MISSING, false);
            return;
        }
        if (license != null) {
            long time = clock.millis();
            boolean active;
            if (license.expiryDate() == BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS) {
                active = true;
            } else {
                // We subtract the grace period from the current time to avoid overflowing on an expiration
                // date that is near Long.MAX_VALUE
                active = time >= license.issueDate() && time - GRACE_PERIOD_DURATION.getMillis() <
license.expiryDate();
            }
            licenseState.update(license.operationMode(), active);

            if (active) {
                if (time < license.expiryDate()) {
                    logger.debug("license [{}] - valid", license.uid());
                } else {
                    logger.warn("license [{}] - grace", license.uid());
                }
            } else {
                logger.warn("license [{}] - expired", license.uid());
            }
        }
    }
```

```
/**
 * Notifies registered licensees of license state change and/or new active license
 * based on the license in <code>currentLicensesMetaData</code>.
 * Additionally schedules license expiry notifications and event callbacks
 * relative to the current license's expiry
 */
private void onUpdate(final LicensesMetaData currentLicensesMetaData) {
    final License license = getLicense(currentLicensesMetaData);
    // license can be null if the trial license is yet to be auto-generated
    // in this case, it is a no-op
    if (license != null) {
        final License previousLicense = currentLicense.get();
        if (license.equals(previousLicense) == false) {
            currentLicense.set(license);
            license.setOperationModeFileWatcher(operationModeFileWatcher);
            scheduler.add(new SchedulerEngine.Job(LICENSE_JOB, nextLicenseCheck(license)));
            for (ExpirationCallback expirationCallback : expirationCallbacks) {
                scheduler.add(new SchedulerEngine.Job(expirationCallback.getId(),
                        (startTime, now) ->
                                expirationCallback.nextScheduledTimeForExpiry(license.expiryDate(), startTime, now)));
            }
            if (previousLicense != null) {
                // remove operationModeFileWatcher to gc the old license object
                previousLicense.removeOperationModeFileWatcher();
            }
            logger.info("license [{}] mode [{}] - valid", license.uid(),
                    license.operationMode().name().toLowerCase(Locale.ROOT));
        }
        updateLicenseState(license);
    }
}

// pkg private for tests
static SchedulerEngine.Schedule nextLicenseCheck(License license) {
    return (startTime, time) -> {
        if (time < license.issueDate()) {
            // when we encounter a license with a future issue date
            // which can happen with autogenerated license,
            // we want to schedule a notification on the license issue date
            // so the license is notificed once it is valid
            // see https://github.com/elastic/x-plugins/issues/983
            return license.issueDate();
        } else if (time < license.expiryDate()) {
            return license.expiryDate();
        } else if (time < license.expiryDate() + GRACE_PERIOD_DURATION.getMillis()) {
            return license.expiryDate() + GRACE_PERIOD_DURATION.getMillis();
        }
        return -1; // license is expired, no need to check again
        };
    }

    public static License getLicense(final MetaData metaData) {
        final LicensesMetaData licensesMetaData = metaData.custom(LicensesMetaData.TYPE);
        return getLicense(licensesMetaData);
    }

    static License getLicense(final LicensesMetaData metaData) {
        if (metaData != null) {
            License license = metaData.getLicense();
            if (license == LicensesMetaData.LICENSE_TOMBSTONE) {
                return license;
            } else if (license != null) {
                boolean autoGeneratedLicense = License.isAutoGeneratedLicense(license.signature());
                if ((autoGeneratedLicense && SelfGeneratedLicense.verify(license))
                        || (!autoGeneratedLicense && LicenseVerifier.verifyLicense(license))) {
                    return license;
                }
            }
        }
        return null;
    }

    private static boolean isProductionMode(Settings settings, DiscoveryNode localNode) {
        final boolean singleNodeDisco = "single-
node".equals(DiscoveryModule.DISCOVERY_TYPE_SETTING.get(settings));
        return singleNodeDisco == false && isBoundToLoopback(localNode) == false;
    }

    private static boolean isBoundToLoopback(DiscoveryNode localNode) {
        return localNode.getAddress().address().getAddress().isLoopbackAddress();
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;
import org.elasticsearch.client.ElasticsearchClient;

public class PostStartTrialAction extends Action<PostStartTrialRequest, PostStartTrialResponse,
PostStartTrialRequestBuilder> {

    public static final PostStartTrialAction INSTANCE = new PostStartTrialAction();
```

```
  public static final String NAME = "cluster:admin/xpack/license/start_trial";

  private PostStartTrialAction() {
    super(NAME);
  }

  @Override
  public PostStartTrialRequestBuilder newRequestBuilder(ElasticsearchClient client) {
    return new PostStartTrialRequestBuilder(client, this);
  }

  @Override
  public PostStartTrialResponse newResponse() {
    return new PostStartTrialResponse();
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.client.ElasticsearchClient;
import org.elasticsearch.rest.action.RestBuilderListener;

public class LicensingClient {

  private final ElasticsearchClient client;

  public LicensingClient(ElasticsearchClient client) {
    this.client = client;
  }

  public PutLicenseRequestBuilder preparePutLicense(License license) {
    return new PutLicenseRequestBuilder(client).setLicense(license);
  }

  public void putLicense(PutLicenseRequest request, ActionListener<PutLicenseResponse> listener) {
    client.execute(PutLicenseAction.INSTANCE, request, listener);
  }

  public GetLicenseRequestBuilder prepareGetLicense() {
    return new GetLicenseRequestBuilder(client);
  }

  public void getLicense(GetLicenseRequest request, ActionListener<GetLicenseResponse> listener) {
    client.execute(GetLicenseAction.INSTANCE, request, listener);
  }

  public DeleteLicenseRequestBuilder prepareDeleteLicense() {
    return new DeleteLicenseRequestBuilder(client);
  }

  public void deleteLicense(DeleteLicenseRequest request, ActionListener<DeleteLicenseResponse> listener) {
    client.execute(DeleteLicenseAction.INSTANCE, request, listener);
  }

  public PostStartTrialRequestBuilder preparePostStartTrial() {
    return new PostStartTrialRequestBuilder(client, PostStartTrialAction.INSTANCE);
  }

  public GetTrialStatusRequestBuilder prepareGetStartTrial() {
    return new GetTrialStatusRequestBuilder(client, GetTrialStatusAction.INSTANCE);
  }

  public void postStartTrial(PostStartTrialRequest request, ActionListener<PostStartTrialResponse> listener) {
    client.execute(PostStartTrialAction.INSTANCE, request, listener);
  }

  public void postStartBasic(PostStartBasicRequest request, ActionListener<PostStartBasicResponse> listener) {
    client.execute(PostStartBasicAction.INSTANCE, request, listener);
  }

  public PostStartBasicRequestBuilder preparePostStartBasic() {
    return new PostStartBasicRequestBuilder(client, PostStartBasicAction.INSTANCE);
  }

  public GetBasicStatusRequestBuilder prepareGetStartBasic() {
    return new GetBasicStatusRequestBuilder(client, GetBasicStatusAction.INSTANCE);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.RestStatus;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.GET;

public class RestGetBasicStatus extends XPackRestHandler {

  RestGetBasicStatus(Settings settings, RestController controller) {
     super(settings);
     controller.registerHandler(GET, URI_BASE + "/license/basic_status", this);
  }

  @Override
  protected RestChannelConsumer doPrepareRequest(RestRequest request, XPackClient client) throws IOException
{
     return channel -> client.licensing().prepareGetStartBasic().execute(
            new RestBuilderListener<GetBasicStatusResponse>(channel) {
               @Override
               public RestResponse buildResponse(GetBasicStatusResponse response, XContentBuilder builder)
throws Exception {
                  builder.startObject();
                  builder.field("eligible_to_start_basic", response.isEligibleToStartBasic());
                  builder.endObject();
                  return new BytesRestResponse(RestStatus.OK, builder);
               }
            });
  }

  @Override
  public String getName() {
     return "xpack_basic_status_action";
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.AcknowledgedRequest;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import java.io.IOException;


public class DeleteLicenseRequest extends AcknowledgedRequest<DeleteLicenseRequest> {

  public DeleteLicenseRequest() {
  }

  @Override
  public ActionRequestValidationException validate() {
     return null;
  }

  @Override
  public void readFrom(StreamInput in) throws IOException {
     super.readFrom(in);
  }

  @Override
  public void writeTo(StreamOutput out) throws IOException {
     super.writeTo(out);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.message.ParameterizedMessage;
import org.apache.logging.log4j.util.Supplier;
import org.elasticsearch.Version;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.logging.Loggers;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.time.Clock;
import java.util.UUID;

public class StartupSelfGeneratedLicenseTask extends ClusterStateUpdateTask {
```

```
/**
 * Max number of nodes licensed by generated trial license
 */
private int selfGeneratedLicenseMaxNodes = 1000;

private final Settings settings;
private final Clock clock;
private final ClusterService clusterService;
private final Logger logger;

public StartupSelfGeneratedLicenseTask(Settings settings, Clock clock, ClusterService clusterService) {
    this.settings = settings;
    this.clock = clock;
    this.clusterService = clusterService;
    this.logger = Loggers.getLogger(getClass(), settings);
}

@Override
public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
    LicensesMetaData licensesMetaData = newState.metaData().custom(LicensesMetaData.TYPE);
    if (logger.isDebugEnabled()) {
        logger.debug("registered self generated license: {}", licensesMetaData);
    }
}

@Override
public ClusterState execute(ClusterState currentState) throws Exception {
    XPackPlugin.checkReadyForXPackCustomMetadata(currentState);
    final MetaData metaData = currentState.metaData();
    final LicensesMetaData currentLicensesMetaData = metaData.custom(LicensesMetaData.TYPE);
    // do not generate a license if any license is present
    if (currentLicensesMetaData == null) {
        String type = LicenseService.SELF_GENERATED_LICENSE_TYPE.get(settings);
        if (SelfGeneratedLicense.validSelfGeneratedType(type) == false) {
            throw new IllegalArgumentException("Illegal self generated license type [" + type +
                    "]. Must be trial or basic.");
        }

        return updateWithLicense(currentState, type);
    } else if (LicenseUtils.licenseNeedsExtended(currentLicensesMetaData.getLicense())) {
        return extendBasic(currentState, currentLicensesMetaData);
    } else {
        return currentState;
    }
}

@Override
```

```
    public void onFailure(String source, @Nullable Exception e) {
        logger.error((Supplier<?>) () -> new ParameterizedMessage("unexpected failure during [{}]", source), e);
    }

    private ClusterState extendBasic(ClusterState currentState, LicensesMetaData currentLicenseMetadata) {
        License license = currentLicenseMetadata.getLicense();
        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
        LicensesMetaData newLicenseMetadata = createBasicLicenseFromExistingLicense(currentLicenseMetadata);
        mdBuilder.putCustom(LicensesMetaData.TYPE, newLicenseMetadata);
        logger.info("Existing basic license has an expiration. Basic licenses no longer expire." +
                "Regenerating license.\n\nOld license:\n {}\n\n New license:\n{}", license,
newLicenseMetadata.getLicense());
        return ClusterState.builder(currentState).metaData(mdBuilder).build();
    }

    private LicensesMetaData createBasicLicenseFromExistingLicense(LicensesMetaData currentLicenseMetadata) {
        License currentLicense = currentLicenseMetadata.getLicense();
        License.Builder specBuilder = License.builder()
                .uid(currentLicense.uid())
                .issuedTo(currentLicense.issuedTo())
                .maxNodes(selfGeneratedLicenseMaxNodes)
                .issueDate(currentLicense.issueDate())
                .type("basic")
                .expiryDate(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS);
        License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder);
        Version trialVersion = currentLicenseMetadata.getMostRecentTrialVersion();
        return new LicensesMetaData(selfGeneratedLicense, trialVersion);
    }

    private ClusterState updateWithLicense(ClusterState currentState, String type) {
        long issueDate = clock.millis();
        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
        long expiryDate;
        if ("basic".equals(type)) {
            expiryDate = LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
        } else {
            expiryDate = issueDate +
LicenseService.NON_BASIC_SELF_GENERATED_LICENSE_DURATION.getMillis();
        }
        License.Builder specBuilder = License.builder()
                .uid(UUID.randomUUID().toString())
                .issuedTo(clusterService.getClusterName().value())
                .maxNodes(selfGeneratedLicenseMaxNodes)
                .issueDate(issueDate)
                .type(type)
                .expiryDate(expiryDate);
        License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder);
        LicensesMetaData licensesMetaData;
        if ("trial".equals(type)) {
            licensesMetaData = new LicensesMetaData(selfGeneratedLicense, Version.CURRENT);
        } else {
            licensesMetaData = new LicensesMetaData(selfGeneratedLicense, null);
        }
        mdBuilder.putCustom(LicensesMetaData.TYPE, licensesMetaData);
        return ClusterState.builder(currentState).metaData(mdBuilder).build();
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.AcknowledgedResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.RestStatus;

import java.io.IOException;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

import static org.elasticsearch.license.PostStartBasicResponse.Status.NEED_ACKNOWLEDGEMENT;

class PostStartBasicResponse extends AcknowledgedResponse {

    private Map<String, String[]> acknowledgeMessages;
    private String acknowledgeMessage;

    enum Status {
        GENERATED_BASIC(true, null, RestStatus.OK),
        ALREADY_USING_BASIC(false, "Operation failed: Current license is basic.", RestStatus.FORBIDDEN),
        NEED_ACKNOWLEDGEMENT(false, "Operation failed: Needs acknowledgement.", RestStatus.OK);

        private final boolean isBasicStarted;
        private final String errorMessage;
        private final RestStatus restStatus;

        Status(boolean isBasicStarted, String errorMessage, RestStatus restStatus) {
            this.isBasicStarted = isBasicStarted;
            this.errorMessage = errorMessage;
            this.restStatus = restStatus;
        }
        boolean isBasicStarted() {
            return isBasicStarted;
        }

        String getErrorMessage() {
            return errorMessage;
        }

        RestStatus getRestStatus() {
            return restStatus;
        }
    }

    private Status status;

    PostStartBasicResponse() {
    }

    PostStartBasicResponse(Status status) {
        this(status, Collections.emptyMap(), null);
    }

    PostStartBasicResponse(Status status, Map<String, String[]> acknowledgeMessages, String
acknowledgeMessage) {
        super(status != NEED_ACKNOWLEDGEMENT);
        this.status = status;
        this.acknowledgeMessages = acknowledgeMessages;
        this.acknowledgeMessage = acknowledgeMessage;
    }

    public Status getStatus() {
        return status;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        super.readFrom(in);
        readAcknowledged(in);
        status = in.readEnum(Status.class);
        acknowledgeMessage = in.readOptionalString();
        int size = in.readVInt();
        Map<String, String[]> acknowledgeMessages = new HashMap<>(size);
        for (int i = 0; i < size; i++) {
            String feature = in.readString();
            int nMessages = in.readVInt();
            String[] messages = new String[nMessages];
            for (int j = 0; j < nMessages; j++) {
```

```
        messages[j] = in.readString();
      }
      acknowledgeMessages.put(feature, messages);
    }
    this.acknowledgeMessages = acknowledgeMessages;
  }

  @Override
  public void writeTo(StreamOutput out) throws IOException {
    super.writeTo(out);
    writeAcknowledged(out);
    out.writeEnum(status);
    out.writeOptionalString(acknowledgeMessage);
    out.writeVInt(acknowledgeMessages.size());
    for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
      out.writeString(entry.getKey());
      out.writeVInt(entry.getValue().length);
      for (String message : entry.getValue()) {
        out.writeString(message);
      }
    }
  }

  @Override
  protected void addCustomFields(XContentBuilder builder, Params params) throws IOException {
    if (status.isBasicStarted()) {
      builder.field("basic_was_started", true);
    } else {
      builder.field("basic_was_started", false);
      builder.field("error_message", status.getErrorMessage());
    }
    if (acknowledgeMessages.isEmpty() == false) {
      builder.startObject("acknowledge");
      builder.field("message", acknowledgeMessage);
      for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
        builder.startArray(entry.getKey());
        for (String message : entry.getValue()) {
          builder.value(message);
        }
        builder.endArray();
      }
      builder.endObject();
    }
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.joda.FormatDateTimeFormatter;
import org.elasticsearch.common.joda.Joda;
import org.joda.time.MutableDateTime;
import org.joda.time.format.DateTimeFormatter;
import org.joda.time.format.ISODateTimeFormat;

public class DateUtils {

  private static final FormatDateTimeFormatter formatDateOnlyFormatter = Joda.forPattern("yyyy-MM-dd");

  private static final DateTimeFormatter dateOnlyFormatter = formatDateOnlyFormatter.parser().withZoneUTC();

  private static final DateTimeFormatter dateTimeFormatter = ISODateTimeFormat.dateTime().withZoneUTC();

  public static long endOfTheDay(String date) {
    try {
      // Try parsing using complete date/time format
      return dateTimeFormatter.parseDateTime(date).getMillis();
    } catch (IllegalArgumentException ex) {
      // Fall back to the date only format
      MutableDateTime dateTime = dateOnlyFormatter.parseMutableDateTime(date);
      dateTime.millisOfDay().set(dateTime.millisOfDay().getMaximumValue());
      return dateTime.getMillis();
    }
  }

  public static long beginningOfTheDay(String date) {
    try {
      // Try parsing using complete date/time format
      return dateTimeFormatter.parseDateTime(date).getMillis();
    } catch (IllegalArgumentException ex) {
      // Fall back to the date only format
      return dateOnlyFormatter.parseDateTime(date).getMillis();
    }

  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```
import org.elasticsearch.action.support.master.AcknowledgedRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

public class DeleteLicenseRequestBuilder extends AcknowledgedRequestBuilder<DeleteLicenseRequest,
DeleteLicenseResponse,
        DeleteLicenseRequestBuilder> {

    public DeleteLicenseRequestBuilder(ElasticsearchClient client) {
        this(client, DeleteLicenseAction.INSTANCE);
    }

    /**
     * Creates new get licenses request builder
     *
     * @param client elasticsearch client
     */
    public DeleteLicenseRequestBuilder(ElasticsearchClient client, DeleteLicenseAction action) {
        super(client, action, new DeleteLicenseRequest());
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

public enum LicensesStatus {
    VALID((byte) 0),
    INVALID((byte) 1),
    EXPIRED((byte) 2);

    private final byte id;

    LicensesStatus(byte id) {
        this.id = id;
    }

    public int id() {
        return id;
    }

    public static LicensesStatus fromId(int id) {
        if (id == 0) {
            return VALID;
        } else if (id == 1) {
            return INVALID;
        } else if (id == 2) {
            return EXPIRED;
        } else {
            throw new IllegalStateException("no valid LicensesStatus for id=" + id);
        }
    }
}
```

/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/

```
/**
* Licensing for xpack.
*
* A {@link org.elasticsearch.license.License} is a signed set of json properties that determine what features
* are available in a running cluster. Licenses are registered through a
* {@link org.elasticsearch.license.PutLicenseRequest}. This action is handled by the master node, which places
* the signed license into the cluster state. Each node listens for cluster state updates via the
* {@link org.elasticsearch.license.LicenseService}, and updates its local copy of the license when it detects
* changes in the cluster state.
*
* The logic for which features are available given the current license is handled by
* {@link org.elasticsearch.license.XPackLicenseState}, which is updated by the
* {@link org.elasticsearch.license.LicenseService} when the license changes.
*/
package org.elasticsearch.license;
```

/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/

```
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeAction;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;
```

```
public class TransportPutLicenseAction extends TransportMasterNodeAction<PutLicenseRequest,
PutLicenseResponse> {

  private final LicenseService licenseService;

  @Inject
  public TransportPutLicenseAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                   LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                                   IndexNameExpressionResolver indexNameExpressionResolver) {
    super(settings, PutLicenseAction.NAME, transportService, clusterService, threadPool, actionFilters,
indexNameExpressionResolver,
        PutLicenseRequest::new);
    this.licenseService = licenseService;
  }

  @Override
  protected String executor() {
    return ThreadPool.Names.MANAGEMENT;
  }

  @Override
  protected PutLicenseResponse newResponse() {
    return new PutLicenseResponse();
  }

  @Override
  protected ClusterBlockException checkBlock(PutLicenseRequest request, ClusterState state) {
    return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_WRITE);
  }

  @Override
  protected void masterOperation(final PutLicenseRequest request, ClusterState state, final
ActionListener<PutLicenseResponse>
      listener) throws ElasticsearchException {
    licenseService.registerLicense(request, listener);
  }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;
```

```
class PostStartBasicRequestBuilder extends ActionRequestBuilder<PostStartBasicRequest,
        PostStartBasicResponse, PostStartBasicRequestBuilder> {

    PostStartBasicRequestBuilder(ElasticsearchClient client, PostStartBasicAction action) {
        super(client, action, new PostStartBasicRequest());
    }

    public PostStartBasicRequestBuilder setAcknowledge(boolean acknowledge) {
        request.acknowledge(acknowledge);
        return this;
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

class GetTrialStatusResponse extends ActionResponse {

    private boolean eligibleToStartTrial;

    GetTrialStatusResponse() {
    }

    GetTrialStatusResponse(boolean eligibleToStartTrial) {
        this.eligibleToStartTrial = eligibleToStartTrial;
    }

    boolean isEligibleToStartTrial() {
        return eligibleToStartTrial;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        eligibleToStartTrial = in.readBoolean();
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        out.writeBoolean(eligibleToStartTrial);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.message.ParameterizedMessage;
import org.elasticsearch.Version;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.time.Clock;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;
import java.util.UUID;
import java.util.concurrent.atomic.AtomicBoolean;
import java.util.concurrent.atomic.AtomicReference;

public class StartTrialClusterTask extends ClusterStateUpdateTask {

    private static final String ACKNOWLEDGEMENT_HEADER = "This API initiates a free 30-day trial for all platinum features. " +
            "By starting this trial, you agree that it is subject to the terms and conditions at" +
            " https://www.elastic.co/legal/trial_license/. To begin your free trial, call /start_trial again and specify " +
            "the \"acknowledge=true\" parameter.";

    private static final Map<String, String[]> ACK_MESSAGES = Collections.singletonMap("security",
            new String[] {"With a trial license, X-Pack security features are available, but are not enabled by default."});

    private final Logger logger;
    private final String clusterName;
    private final PostStartTrialRequest request;
    private final ActionListener<PostStartTrialResponse> listener;
    private final Clock clock;

    StartTrialClusterTask(Logger logger, String clusterName, Clock clock, PostStartTrialRequest request,
                                ActionListener<PostStartTrialResponse> listener) {
        this.logger = logger;
        this.clusterName = clusterName;
        this.request = request;
        this.listener = listener;
        this.clock = clock;
    }

    @Override
    public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
        LicensesMetaData oldLicensesMetaData = oldState.metaData().custom(LicensesMetaData.TYPE);
        logger.debug("started self generated trial license: {}", oldLicensesMetaData);

        if (request.isAcknowledged() == false) {
            listener.onResponse(new
PostStartTrialResponse(PostStartTrialResponse.Status.NEED_ACKNOWLEDGEMENT,
                    ACK_MESSAGES, ACKNOWLEDGEMENT_HEADER));
        } else if (oldLicensesMetaData == null || oldLicensesMetaData.isEligibleForTrial()) {
            listener.onResponse(new PostStartTrialResponse(PostStartTrialResponse.Status.UPGRADED_TO_TRIAL));
        } else {
            listener.onResponse(new
PostStartTrialResponse(PostStartTrialResponse.Status.TRIAL_ALREADY_ACTIVATED));
        }
    }

    @Override
    public ClusterState execute(ClusterState currentState) throws Exception {
        XPackPlugin.checkReadyForXPackCustomMetadata(currentState);
        LicensesMetaData currentLicensesMetaData = currentState.metaData().custom(LicensesMetaData.TYPE);

        if (request.isAcknowledged() == false) {
            return currentState;
        } else if (currentLicensesMetaData == null || currentLicensesMetaData.isEligibleForTrial()) {
            long issueDate = clock.millis();
            MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
            long expiryDate = issueDate +
LicenseService.NON_BASIC_SELF_GENERATED_LICENSE_DURATION.getMillis();

            License.Builder specBuilder = License.builder()
                    .uid(UUID.randomUUID().toString())
                    .issuedTo(clusterName)
                    .maxNodes(LicenseService.SELF_GENERATED_LICENSE_MAX_NODES)
                    .issueDate(issueDate)
                    .type(request.getType())
                    .expiryDate(expiryDate);
            License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder);
            LicensesMetaData newLicensesMetaData = new LicensesMetaData(selfGeneratedLicense,
Version.CURRENT);
            mdBuilder.putCustom(LicensesMetaData.TYPE, newLicensesMetaData);
            return ClusterState.builder(currentState).metaData(mdBuilder).build();
        } else {
            return currentState;
        }
    }

    @Override
    public void onFailure(String source, @Nullable Exception e) {
        logger.error(new ParameterizedMessage("unexpected failure during [{}]", source), e);
        listener.onFailure(e);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.AcknowledgedRequest;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

public class PostStartBasicRequest extends AcknowledgedRequest<PostStartBasicRequest> {

    private boolean acknowledge = false;

    @Override
    public ActionRequestValidationException validate() {
        return null;
    }

    public PostStartBasicRequest acknowledge(boolean acknowledge) {
        this.acknowledge = acknowledge;
        return this;
    }

    public boolean isAcknowledged() {
        return acknowledge;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        super.readFrom(in);
        acknowledge = in.readBoolean();
    }
```

```
  @Override
  public void writeTo(StreamOutput out) throws IOException {
    super.writeTo(out);
    out.writeBoolean(acknowledge);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

class GetBasicStatusResponse extends ActionResponse {

  private boolean eligibleToStartBasic;

  GetBasicStatusResponse() {
  }

  GetBasicStatusResponse(boolean eligibleToStartBasic) {
    this.eligibleToStartBasic = eligibleToStartBasic;
  }

  boolean isEligibleToStartBasic() {
    return eligibleToStartBasic;
  }

  @Override
  public void readFrom(StreamInput in) throws IOException {
    eligibleToStartBasic = in.readBoolean();
  }

  @Override
  public void writeTo(StreamOutput out) throws IOException {
    out.writeBoolean(eligibleToStartBasic);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.logging.LoggerMessageFormat;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.xpack.core.scheduler.SchedulerEngine;

import java.util.UUID;

abstract class ExpirationCallback {

  static final String EXPIRATION_JOB_PREFIX = ".license_expiration_job_";

  public enum Orientation {PRE, POST}

  /**
   * Callback that is triggered every <code>frequency</code> when
   * current time is between <code>max</code> and <code>min</code>
   * before license expiry.
   */
  public abstract static class Pre extends ExpirationCallback {

    /**
     * Callback schedule prior to license expiry
     *
     * @param min       latest relative time to execute before license expiry
     * @param max       earliest relative time to execute before license expiry
     * @param frequency interval between execution
     */
    Pre(TimeValue min, TimeValue max, TimeValue frequency) {
      super(Orientation.PRE, min, max, frequency);
    }
  }

  /**
   * Callback that is triggered every <code>frequency</code> when
   * current time is between <code>min</code> and <code>max</code>
   * after license expiry.
   */
  public abstract static class Post extends ExpirationCallback {

    /**
     * Callback schedule after license expiry
     *
     * @param min       earliest relative time to execute after license expiry
     * @param max       latest relative time to execute after license expiry
     * @param frequency interval between execution
```

```
         */
        Post(TimeValue min, TimeValue max, TimeValue frequency) {
            super(Orientation.POST, min, max, frequency);
        }
    }

    private final String id;
    private final Orientation orientation;
    private final long min;
    private final long max;
    private final long frequency;

    private ExpirationCallback(Orientation orientation, TimeValue min, TimeValue max, TimeValue frequency) {
        this.orientation = orientation;
        this.min = (min == null) ? 0 : min.getMillis();
        this.max = (max == null) ? Long.MAX_VALUE : max.getMillis();
        this.frequency = frequency.getMillis();
        this.id = String.join("", EXPIRATION_JOB_PREFIX, UUID.randomUUID().toString());
    }

    public final String getId() {
        return id;
    }

    public final long getFrequency() {
        return frequency;
    }

    /**
     * Calculates the delay for the next trigger time. When <code>now</code> is in a
     * valid time bracket with respect to <code>expirationDate</code>, the delay is 0.
     * When <code>now</code> is before the time bracket, than delay to the start of the
     * time bracket and when <code>now</code> is passed the valid time bracket, the delay
     * is <code>null</code>
     * @param expirationDate license expiry date in milliseconds
     * @param now current time in milliseconds
     * @return time delay
     */
    final TimeValue delay(long expirationDate, long now) {
        final TimeValue delay;
        switch (orientation) {
            case PRE:
                if (expirationDate >= now) {
                    // license not yet expired
                    long preExpiryDuration = expirationDate - now;
                    if (preExpiryDuration > max) {
                        // license duration is longer than maximum duration, delay it to the first match time
                        delay = TimeValue.timeValueMillis(preExpiryDuration - max);
                    } else if (preExpiryDuration <= max && preExpiryDuration >= min) {
                        // no delay in valid time bracket
                        delay = TimeValue.timeValueMillis(0);
                    } else {
                        // passed last match time
                        delay = null;
                    }
                } else {
                    // invalid after license expiry
                    delay = null;
                }
                break;
            case POST:
                if (expirationDate >= now) {
                    // license not yet expired, delay it to the first match time
                    delay = TimeValue.timeValueMillis(expirationDate - now + min);
                } else {
                    // license has expired
                    long expiredDuration = now - expirationDate;
                    if (expiredDuration < min) {
                        // license expiry duration is shorter than minimum duration, delay it to the first match time
                        delay = TimeValue.timeValueMillis(min - expiredDuration);
                    } else if (expiredDuration >= min && expiredDuration <= max) {
                        // no delay in valid time bracket
                        delay = TimeValue.timeValueMillis(0);
                    } else {
                        // passed last match time
                        delay = null;
                    }
                }
                break;
            default:
                throw new IllegalStateException("orientation [" + orientation + "] unknown");
        }
        return delay;
    }

    /**
     * {@link SchedulerEngine.Schedule#nextScheduledTimeAfter(long, long)} with respect to
     * license expiry date
     */
    public final long nextScheduledTimeForExpiry(long expiryDate, long startTime, long time) {
        TimeValue delay = delay(expiryDate, time);
        if (delay != null) {
            long delayInMillis = delay.getMillis();
            if (delayInMillis == 0L) {
                if (startTime == time) {
                    // initial trigger and in time bracket, schedule immediately
```

```
                    return time;
                } else {
                    // in time bracket, add frequency
                    return time + frequency;
                }
            } else {
                // not in time bracket
                return time + delayInMillis;
            }
        }
        return -1;
    }

    /**
     * Code to execute when the expiry callback is triggered in a valid
     * time bracket
     * @param license license to operate on
     */
    public abstract void on(License license);

    public final String toString() {
        return LoggerMessageFormat.format(null, "ExpirationCallback:(orientation [{}],  min [{}], max [{}], freq
[{}])",
                orientation.name(), TimeValue.timeValueMillis(min), TimeValue.timeValueMillis(max),
                TimeValue.timeValueMillis(frequency));
    }
}
{
  "metadata": {
    "name": "X-Pack Monitoring: License Expiration",
    "xpack": {
      "link": "license",
      "expires_days": [ 60, 30, 14, 7 ],
      "severity": 0,
      "alert_index": ".monitoring-alerts-6",
      "cluster_uuid": "${monitoring.watch.cluster_uuid}",
      "type": "monitoring",
      "version_created": 6020099,
      "watch": "${monitoring.watch.id}"
    }
  },
  "trigger": {
    "schedule": {
      "interval": "1m"
    }
  },
  "input": {
    "chain": {
```

```
"inputs": [
  {
    "check": {
      "search": {
        "request": {
          "indices": [
            ".monitoring-es-*"
          ],
          "body": {
            "size": 1,
            "sort": [
              {
                "timestamp": {
                  "order": "desc"
                }
              }
            ],
            "_source": [
              "license.*"
            ],
            "query": {
              "bool": {
                "filter": [
                  {
                    "term": {
                      "cluster_uuid": "{{ctx.metadata.xpack.cluster_uuid}}"
                    }
                  },
                  {
                    "term": {
                      "type": "cluster_stats"
                    }
                  },
                  {
                    "range": {
                      "timestamp": {
                        "gte": "now-2m"
                      }
                    }
                  }
                ]
              }
            }
          }
        }
      }
    }
  },
  {
    "alert": {
      "search": {
        "request": {
          "indices": [
            ".monitoring-alerts-6"
          ],
          "body": {
            "size": 1,
            "terminate_after": 1,
            "query": {
              "bool": {
                "filter": {
                  "term": {
                    "_id": "{{ctx.watch_id}}"
                  }
                }
              }
            },
            "sort": [
              { "timestamp": { "order": "desc" } }
            ]
          }
        }
      }
    }
  },
  {
    "kibana_settings": {
      "search": {
        "request": {
          "indices": [
            ".monitoring-kibana-6-*"
          ],
          "body": {
            "size": 1,
            "query": {
              "bool": {
                "filter": {
                  "term": {
                    "type": "kibana_settings"
                  }
                }
              }
            },
            "sort": [
              {
                "timestamp": {
```

```
                       "order": "desc"
                     }
                   }
                 ]
               }
             }
           }
         }
       }
     ]
   }
 },
 "condition": {
   "script": {
     "source": "if (ctx.payload.check.hits.total == 0) {return false;}def license =
ctx.payload.check.hits.hits[0]._source.license;if (license == null) {return false;}ctx.vars.fails_check = false;Instant
expiry = Instant.ofEpochMilli(license.expiry_date_in_millis);ctx.vars.expiry = expiry;if (license.status != 'active')
{ctx.vars.expired = true;ctx.vars.fails_check = true;ctx.metadata.xpack.severity = 2001;} else {Instant now =
Instant.ofEpochMilli(new Date().getTime());ctx.vars.now = now;for (int i = ctx.metadata.xpack.expires_days.length
- 1;i > -1;--i) {if (license.type == 'trial' && i < 2) {break;}Instant fromNow =
now.plusSeconds(ctx.metadata.xpack.expires_days[i] * 24 * 60 * 60);if (fromNow.isAfter(expiry))
{ctx.vars.fails_check = true;ctx.metadata.xpack.severity = i * 1000;break;}}}ctx.vars.not_resolved =
(ctx.payload.alert.hits.total == 1 && ctx.payload.alert.hits.hits[0]._source.resolved_timestamp ==
null);ctx.vars.update = ctx.vars.fails_check || ctx.vars.not_resolved;"
   }
 },
 "transform": {
   "script": {
     "source": "ctx.vars.email_recipient = (ctx.payload.kibana_settings.hits.total > 0) ?
ctx.payload.kibana_settings.hits.hits[0]._source.kibana_settings.xpack.default_admin_email : null;ctx.vars.is_new =
ctx.vars.fails_check && !ctx.vars.not_resolved;ctx.vars.is_resolved = !ctx.vars.fails_check &&
ctx.vars.not_resolved;def alertMessage = null;if (ctx.vars.fails_check) { alertMessage = 'Update your license.';} if
(ctx.vars.not_resolved) { ctx.payload = ctx.payload.alert.hits.hits[0]._source;ctx.payload.metadata =
ctx.metadata.xpack;if (ctx.vars.fails_check == false) { ctx.payload.resolved_timestamp = ctx.execution_time;} }
else { ctx.payload = [ 'timestamp': ctx.execution_time, 'prefix': 'This cluster\\'s license is going to expire in
{{#relativeTime}}metadata.time{{/relativeTime}} at {{#absoluteTime}}metadata.time{{/absoluteTime}}.',
'message': alertMessage, 'metadata': ctx.metadata.xpack ];} if (ctx.vars.fails_check) { ctx.payload.metadata.time =
ctx.vars.expiry.toString();} ctx.payload.update_timestamp = ctx.execution_time;return ctx.payload;"
   }
 },
 "actions": {
   "add_to_alerts_index": {
     "index": {
       "index": ".monitoring-alerts-6",
       "doc_type": "doc",
       "doc_id": "${monitoring.watch.unique_id}"
     }
   },
   "send_email_to_admin": {
     "condition": {
       "script": "return ctx.vars.email_recipient != null && (ctx.vars.is_new || ctx.vars.is_resolved)"
     },
     "email": {
       "to": "X-Pack Admin <{{ctx.vars.email_recipient}}>",
       "from": "X-Pack Admin <{{ctx.vars.email_recipient}}>",
       "subject":
"[{{#ctx.vars.is_new}}NEW{{/ctx.vars.is_new}}{{#ctx.vars.is_resolved}}RESOLVED{{/ctx.vars.is_resolved}}]
{{ctx.metadata.name}}",
       "body": {
         "text": "{{#ctx.vars.is_resolved}}This cluster alert has been resolved: {{/ctx.vars.is_resolved}} This cluster's
license {{#ctx.vars.is_new}}is going to expire on
{{ctx.payload.metadata.time}}{{/ctx.vars.is_new}}{{#ctx.vars.is_resolved}}was going to
expire{{ctx.payload.metadata.time}}{{/ctx.vars.is_resolved}}.
{{#ctx.vars.is_new}}{{ctx.payload.message}}{{/ctx.vars.is_new}}"
       }
     }
   }
 }
}
/*
*                                 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 Kasper B. Graversen
 *
 *   Licensed under the Apache License, Version 2.0 (the "License");
 *   you may not use this file except in compliance with the License.
 *   You may obtain a copy of the License at
 *
 *       http://www.apache.org/licenses/LICENSE-2.0
 *
 *   Unless required by applicable law or agreed to in writing, software
 *   distributed under the License is distributed on an "AS IS" BASIS,
 *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *   See the License for the specific language governing permissions and
 *   limitations under the License.
 */
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.action.support.PlainActionFuture;
import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.rest.RestStatus;
import org.elasticsearch.test.junit.annotations.TestLogging;
import org.elasticsearch.transport.Transport;
import org.elasticsearch.xpack.core.TestXPackTransportClient;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.ml.action.CloseJobAction;
import org.elasticsearch.xpack.core.ml.action.DeleteDatafeedAction;
import org.elasticsearch.xpack.core.ml.action.DeleteJobAction;
import org.elasticsearch.xpack.core.ml.action.GetDatafeedsStatsAction;
import org.elasticsearch.xpack.core.ml.action.GetJobsStatsAction;
import org.elasticsearch.xpack.core.ml.action.OpenJobAction;
import org.elasticsearch.xpack.core.ml.action.PutDatafeedAction;
import org.elasticsearch.xpack.core.ml.action.PutJobAction;
import org.elasticsearch.xpack.core.ml.action.StartDatafeedAction;
import org.elasticsearch.xpack.core.ml.action.StopDatafeedAction;
import org.elasticsearch.xpack.core.ml.client.MachineLearningClient;
import org.elasticsearch.xpack.core.ml.datafeed.DatafeedState;
import org.elasticsearch.xpack.core.ml.job.config.JobState;
import org.elasticsearch.persistent.PersistentTasksCustomMetaData;
import org.elasticsearch.xpack.ml.LocalStateMachineLearning;
import org.elasticsearch.xpack.ml.support.BaseMlIntegTestCase;
import org.junit.Before;

import java.util.Collections;

import static org.hamcrest.Matchers.containsString;
import static org.hamcrest.Matchers.hasItem;
import static org.hamcrest.Matchers.is;

@TestLogging("org.elasticsearch.xpack.ml.action:DEBUG")
public class MachineLearningLicensingTests extends BaseMlIntegTestCase {

  @Before
  public void resetLicensing() {
    enableLicensing();

    ensureStableCluster(1);
    ensureYellow();
  }

  public void testMachineLearningPutJobActionRestricted() throws Exception {
    String jobId = "testmachinelearningputjobactionrestricted";
    // Pick a license that does not allow machine learning
    License.OperationMode mode = randomInvalidLicenseType();
    enableLicensing(mode);
    assertMLAllowed(false);
    // test that license restricted apis do not work
    Settings settings = internalCluster().transportClient().settings();
    try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
```

```
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> listener = PlainActionFuture. newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), listener);
            listener.actionGet();
            fail("put job action should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
            assertThat(e.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.MACHINE_LEARNING));
        }

        // Pick a license that does allow machine learning
        mode = randomValidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(true);
        // test that license restricted apis do now work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), listener);
            PutJobAction.Response response = listener.actionGet();
            assertNotNull(response);
        }
    }

    public void testMachineLearningOpenJobActionRestricted() throws Exception {
        String jobId = "testmachinelearningopenjobactionrestricted";
        assertMLAllowed(true);
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response response = putJobListener.actionGet();
            assertNotNull(response);
        }

        // Pick a license that does not allow machine learning
        License.OperationMode mode = randomInvalidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(false);
        // test that license restricted apis do not work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
```

```
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<OpenJobAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), listener);
            listener.actionGet();
            fail("open job action should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
            assertThat(e.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.MACHINE_LEARNING));
        }

        // Pick a license that does allow machine learning
        mode = randomValidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(true);

        // now that the license is invalid, the job should get closed:
        assertBusy(() -> {
            JobState jobState = getJobStats(jobId).getState();
            assertEquals(JobState.CLOSED, jobState);
        });

        // test that license restricted apis do now work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<OpenJobAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), listener);
            OpenJobAction.Response response = listener.actionGet();
            assertNotNull(response);
        }
    }

    public void testMachineLearningPutDatafeedActionRestricted() throws Exception {
        String jobId = "testmachinelearningputdatafeedactionrestricted";
        String datafeedId = jobId + "-datafeed";
        assertMLAllowed(true);
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
        }

        // Pick a license that does not allow machine learning
        License.OperationMode mode = randomInvalidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(false);
        // test that license restricted apis do not work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putDatafeed(
                    new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId, Collections.singletonList(jobId))),
listener);
            listener.actionGet();
            fail("put datafeed action should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
            assertThat(e.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.MACHINE_LEARNING));
        }

        // Pick a license that does allow machine learning
        mode = randomValidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(true);
        // test that license restricted apis do now work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putDatafeed(
                    new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId, Collections.singletonList(jobId))),
listener);
            PutDatafeedAction.Response response = listener.actionGet();
            assertNotNull(response);
        }
    }

    public void testAutoCloseJobWithDatafeed() throws Exception {
        String jobId = "testautoclosejobwithdatafeed";
        String datafeedId = jobId + "-datafeed";
        assertMLAllowed(true);
        String datafeedIndex = jobId + "-data";
        prepareCreate(datafeedIndex).addMapping("type", "{\"type\":{\"properties\":{\"time\":{\"type\":\"date\"}}}}",
            XContentType.JSON).get();
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            // put job
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
            // put datafeed
            PlainActionFuture<PutDatafeedAction.Response> putDatafeedListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putDatafeed(
                    new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId,
                            Collections.singletonList(datafeedIndex))), putDatafeedListener);
            PutDatafeedAction.Response putDatafeedResponse = putDatafeedListener.actionGet();
            assertNotNull(putDatafeedResponse);
            // open job
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);
            // start datafeed
            PlainActionFuture<StartDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(new StartDatafeedAction.Request(datafeedId, 0L),
listener);
            listener.actionGet();
        }

        if (randomBoolean()) {
            enableLicensing(randomInvalidLicenseType());
        } else {
            disableLicensing();
        }
        assertMLAllowed(false);

        // now that the license is invalid, the job should be closed and datafeed stopped:
        assertBusy(() -> {
            JobState jobState = getJobStats(jobId).getState();
            assertEquals(JobState.CLOSED, jobState);

            DatafeedState datafeedState = getDatafeedStats(datafeedId).getDatafeedState();
            assertEquals(DatafeedState.STOPPED, datafeedState);

            ClusterState state = client().admin().cluster().prepareState().get().getState();
            PersistentTasksCustomMetaData tasks = state.metaData().custom(PersistentTasksCustomMetaData.TYPE);
            assertEquals(0, tasks.taskMap().size());
        });

        enableLicensing(randomValidLicenseType());
        assertMLAllowed(true);

        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
    client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
    ());
            // open job
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);
            // start datafeed
            PlainActionFuture<StartDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(new StartDatafeedAction.Request(datafeedId, 0L),
    listener);
            listener.actionGet();
        }

        assertBusy(() -> {
            JobState jobState = getJobStats(jobId).getState();
            assertEquals(JobState.OPENED, jobState);

            DatafeedState datafeedState = getDatafeedStats(datafeedId).getDatafeedState();
            assertEquals(DatafeedState.STARTED, datafeedState);

            ClusterState state = client().admin().cluster().prepareState().get().getState();
            PersistentTasksCustomMetaData tasks = state.metaData().custom(PersistentTasksCustomMetaData.TYPE);
            assertEquals(2, tasks.taskMap().size());
        });

        if (randomBoolean()) {
            enableLicensing(randomInvalidLicenseType());
        } else {
            disableLicensing();
        }
        assertMLAllowed(false);

        // now that the license is invalid, the job should be closed and datafeed stopped:
        assertBusy(() -> {
            JobState jobState = getJobStats(jobId).getState();
            assertEquals(JobState.CLOSED, jobState);

            DatafeedState datafeedState = getDatafeedStats(datafeedId).getDatafeedState();
            assertEquals(DatafeedState.STOPPED, datafeedState);

            ClusterState state = client().admin().cluster().prepareState().get().getState();
            PersistentTasksCustomMetaData tasks = state.metaData().custom(PersistentTasksCustomMetaData.TYPE);
            assertEquals(0, tasks.taskMap().size());
        });
```

```
  }

  public void testMachineLearningStartDatafeedActionRestricted() throws Exception {
    String jobId = "testmachinelearningstartdatafeedactionrestricted";
    String datafeedId = jobId + "-datafeed";
    assertMLAllowed(true);
    String datafeedIndex = jobId + "-data";
    prepareCreate(datafeedIndex).addMapping("type", "{\"type\":{\"properties\":{\"time\":{\"type\":\"date\"}}}}",
      XContentType.JSON).get();
    // test that license restricted apis do now work
    Settings settings = internalCluster().transportClient().settings();
    try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
      PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
      new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
      PutJobAction.Response putJobResponse = putJobListener.actionGet();
      assertNotNull(putJobResponse);
      PlainActionFuture<PutDatafeedAction.Response> putDatafeedListener = PlainActionFuture.newFuture();
      new MachineLearningClient(client).putDatafeed(
          new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId,
              Collections.singletonList(datafeedIndex))), putDatafeedListener);
      PutDatafeedAction.Response putDatafeedResponse = putDatafeedListener.actionGet();
      assertNotNull(putDatafeedResponse);
      PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
      new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
      OpenJobAction.Response openJobResponse = openJobListener.actionGet();
      assertNotNull(openJobResponse);
    }

    // Pick a license that does not allow machine learning
    License.OperationMode mode = randomInvalidLicenseType();
    enableLicensing(mode);
    assertMLAllowed(false);

    // now that the license is invalid, the job should get closed:
    assertBusy(() -> {
      JobState jobState = getJobStats(jobId).getState();
      assertEquals(JobState.CLOSED, jobState);
      ClusterState state = client().admin().cluster().prepareState().get().getState();
      PersistentTasksCustomMetaData tasks = state.metaData().custom(PersistentTasksCustomMetaData.TYPE);
      assertEquals(0, tasks.taskMap().size());
    });

    // test that license restricted apis do not work
    try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
```

```
            PlainActionFuture<StartDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(new StartDatafeedAction.Request(datafeedId, 0L),
listener);
            listener.actionGet();
            fail("start datafeed action should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
            assertThat(e.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.MACHINE_LEARNING));
        }

        // Pick a license that does allow machine learning
        mode = randomValidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(true);
        // test that license restricted apis do now work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            // re-open job now that the license is valid again
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);

            PlainActionFuture<StartDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(new StartDatafeedAction.Request(datafeedId, 0L),
listener);
            StartDatafeedAction.Response response = listener.actionGet();
            assertNotNull(response);
        }
    }

    public void testMachineLearningStopDatafeedActionNotRestricted() throws Exception {
        String jobId = "testmachinelearningstopdatafeedactionnotrestricted";
        String datafeedId = jobId + "-datafeed";
        assertMLAllowed(true);
        String datafeedIndex = jobId + "-data";
        prepareCreate(datafeedIndex).addMapping("type", "{\"type\":{\"properties\":{\"time\":{\"type\":\"date\"}}}}",
            XContentType.JSON).get();
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
            PlainActionFuture<PutDatafeedAction.Response> putDatafeedListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putDatafeed(
                    new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId,
                            Collections.singletonList(datafeedIndex))), putDatafeedListener);
            PutDatafeedAction.Response putDatafeedResponse = putDatafeedListener.actionGet();
            assertNotNull(putDatafeedResponse);
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);
            PlainActionFuture<StartDatafeedAction.Response> startDatafeedListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(
                new StartDatafeedAction.Request(datafeedId, 0L), startDatafeedListener);
            StartDatafeedAction.Response startDatafeedResponse = startDatafeedListener.actionGet();
            assertNotNull(startDatafeedResponse);
        }

        boolean invalidLicense = randomBoolean();
        if (invalidLicense) {
            enableLicensing(randomInvalidLicenseType());
        } else {
            enableLicensing(randomValidLicenseType());
        }

        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<StopDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).stopDatafeed(new StopDatafeedAction.Request(datafeedId), listener);
            if (invalidLicense) {
                // the stop datafeed due to invalid license happens async, so check if the datafeed turns into stopped state:
                assertBusy(() -> {
                    GetDatafeedsStatsAction.Response response =
                            new MachineLearningClient(client)
                                    .getDatafeedsStats(new GetDatafeedsStatsAction.Request(datafeedId)).actionGet();
                    assertEquals(DatafeedState.STOPPED, response.getResponse().results().get(0).getDatafeedState());
                });
            } else {
                listener.actionGet();
            }

            if (invalidLicense) {
                // the close due to invalid license happens async, so check if the job turns into closed state:
                assertBusy(() -> {
                    GetJobsStatsAction.Response response =
                            new MachineLearningClient(client).getJobsStats(new
GetJobsStatsAction.Request(jobId)).actionGet();
                assertEquals(JobState.CLOSED, response.getResponse().results().get(0).getState());
            });
        }
      }
  }

  public void testMachineLearningCloseJobActionNotRestricted() throws Exception {
      String jobId = "testmachinelearningclosejobactionnotrestricted";
      assertMLAllowed(true);
      // test that license restricted apis do now work
      Settings settings = internalCluster().transportClient().settings();
      try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
          PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
          new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
          PutJobAction.Response putJobResponse = putJobListener.actionGet();
          assertNotNull(putJobResponse);
          PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
          new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
          OpenJobAction.Response openJobResponse = openJobListener.actionGet();
          assertNotNull(openJobResponse);
      }

      boolean invalidLicense = randomBoolean();
      if (invalidLicense) {
          enableLicensing(randomInvalidLicenseType());
      } else {
          enableLicensing(randomValidLicenseType());
      }

      try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
          PlainActionFuture<CloseJobAction.Response> listener = PlainActionFuture.newFuture();
          CloseJobAction.Request request = new CloseJobAction.Request(jobId);
          request.setCloseTimeout(TimeValue.timeValueSeconds(20));
          if (invalidLicense) {
              // the close due to invalid license happens async, so check if the job turns into closed state:
              assertBusy(() -> {
                  GetJobsStatsAction.Response response =
                          new MachineLearningClient(client).getJobsStats(new
GetJobsStatsAction.Request(jobId)).actionGet();
                  assertEquals(JobState.CLOSED, response.getResponse().results().get(0).getState());
              });
          } else {
              new MachineLearningClient(client).closeJob(request, listener);
```

```
                listener.actionGet();
            }
        }
    }

    public void testMachineLearningDeleteJobActionNotRestricted() throws Exception {
        String jobId = "testmachinelearningclosejobactionnotrestricted";
        assertMLAllowed(true);
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
        }

        // Pick a random license
        License.OperationMode mode = randomLicenseType();
        enableLicensing(mode);

        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<DeleteJobAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).deleteJob(new DeleteJobAction.Request(jobId), listener);
            listener.actionGet();
        }
    }

    public void testMachineLearningDeleteDatafeedActionNotRestricted() throws Exception {
        String jobId = "testmachinelearningdeletedatafeedactionnotrestricted";
        String datafeedId = jobId + "-datafeed";
        assertMLAllowed(true);
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
            PlainActionFuture<PutDatafeedAction.Response> putDatafeedListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putDatafeed(
                    new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId,
                    Collections.singletonList(jobId))), putDatafeedListener);
            PutDatafeedAction.Response putDatafeedResponse = putDatafeedListener.actionGet();
            assertNotNull(putDatafeedResponse);
        }

        // Pick a random license
        License.OperationMode mode = randomLicenseType();
        enableLicensing(mode);

        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<DeleteDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).deleteDatafeed(new DeleteDatafeedAction.Request(datafeedId),
listener);
            listener.actionGet();
        }
    }

    private static OperationMode randomInvalidLicenseType() {
        return randomFrom(License.OperationMode.GOLD, License.OperationMode.STANDARD,
License.OperationMode.BASIC);
    }

    private static OperationMode randomValidLicenseType() {
        return randomFrom(License.OperationMode.TRIAL, License.OperationMode.PLATINUM);
    }

    private static OperationMode randomLicenseType() {
        return randomFrom(License.OperationMode.values());
    }

    private static void assertMLAllowed(boolean expected) {
        for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
            assertEquals(licenseState.isMachineLearningAllowed(), expected);
        }
    }

    public static void disableLicensing() {
        disableLicensing(randomValidLicenseType());
    }

    public static void disableLicensing(License.OperationMode operationMode) {
        for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
            licenseState.update(operationMode, false);
        }
    }
    public static void enableLicensing() {
        enableLicensing(randomValidLicenseType());
    }

    public static void enableLicensing(License.OperationMode operationMode) {
        for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
            licenseState.update(operationMode, true);
        }
    }
}
---
teardown:
 - do:
     xpack.license.post:
       acknowledge: true
       body: |
         {"licenses":[{"uid":"894371dc-9t49-4997-93cb-
8o2e3r7fa6a8","type":"trial","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1916956799999,"max
_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"AAAAAgAAAA0FWh0T9njItjQ2qammAAABmC
9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXM
vRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0ppTHBkWkprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095M
XRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEphZH
BEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQa
VB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c28zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2VE1q
bVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9wTTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldMbmMzSGtQ
c3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdENGWFRreXNKNkE9PQAAAQBZhvozA0
trrxhUZ1QbaTsKTna9C5KVQ6pv8yg1pnsBpZXCl8kX1SrgoFn1bXq61IvJwfw5qnmYNiH3hRhTO9EyaCBqaLk8
NXZQ6TrRkQSpEnnBwAYUkZeKXsIuBoOk4B4mzwC/r8aMAkzrTiEBtBbog+57cSaU9y37Gkdd+1jXCQrxP+jO
EUf7gnXWZvE6oeRroLvCt1fYn09k0CF8kKTbrPTSjC6igZR3uvTHyee74XQ9PRavvHax73T4UOEdQZX/P1ibS
QIWKbBRD5YQ1POYVjTayoltTnWLMxfEcAkkATJZLhpBEHST7kZWjrTS6J1dCReJc7a8Vsj/78HXvOIy"}]}
---
"Installing and getting license works":

 ## current license version
 - do:
     xpack.license.post:
       acknowledge: true
       body: |
         {"licenses":[{"uid":"894371dc-9t49-4997-93cb-
8o2e3r7fa6a8","type":"trial","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1916956799999,"max
_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"AAAAAgAAAA0FWh0T9njItjQ2qammAAABmC
9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXM
vRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0ppTHBkWkprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095M
XRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEphZH
BEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQa
VB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c28zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2VE1q
bVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9wTTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldMbmMzSGtQ
c3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdENGWFRreXNKNkE9PQAAAQBZhvozA0
```

```
trrxhUZ1QbaTsKTna9C5KVQ6pv8yg1pnsBpZXCl8kX1SrgoFn1bXq61IvJwfw5qnmYNiH3hRhTO9EyaCBqaLk8
NXZQ6TrRkQSpEnnBwAYUkZeKXsIuBoOk4B4mzwC/r8aMAkzrTiEBtBbog+57cSaU9y37Gkdd+1jXCQrxP+jO
EUf7gnXWZvE6oeRroLvCt1fYn09k0CF8kKTbrPTSjC6igZR3uvTHyee74XQ9PRavvHax73T4UOEdQZX/P1ibS
QIWKbBRD5YQ1POYVjTayoltTnWLMxfEcAkkATJZLhpBEHST7kZWjrTS6J1dCReJc7a8Vsj/78HXvOIy"}]}

 - match: { license_status:  "valid" }

 - do:
     xpack.license.get: {}

 ## a license object has 11 attributes
 - length: { license: 11 }

 ## bwc for licenses format
 - do:
     xpack.license.post:
       acknowledge: true
       body: |
         {"licenses":[{"uid":"893361dc-9749-4997-93cb-
802e3d7fa4a8","type":"gold","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1914278399999,"ma
x_nodes":1,"issued_to":"issued_to","issuer":"issuer","signature":"AAAAAwAAAA2T3vqdBBetKQaBgxipAAAB
mC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2M
XMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0ppTHBkWkprZWZSQi9iNmRQNkw1SlpLN0lDalZCS09
5MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEph
ZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOE
hQaVB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c28zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2V
E1qbVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9wTTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldMbmMzS
GtQc3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdENGWFRreXNKNkE9PQAAAQB7pG
FYgawfLm9zzT80LvcLHjy1t/v2uSzCQWKdXXhrwSy4WrAH2uK/+PEiQ7aEpW5erLsyJ5KLA6OEZJDaP7r+mj
OPuLt0++l5j4DMn7ybMzOPHXWBc6LETE3+pp0GZPyOmwsDkZSRUegTtciR2R6z+mdnGrhOYM80y08KVW
whdU/DHw41MK7ePo6tq73Nz49y9lDgt9fxA0t4ggEBPbnTDDBVQ25AjauY8sa0M5eg9rDDRayw1KamYWrara8
PIGX+2YjhtUeQhmlCPdlxc9wECJ7/knPss5bI3ZoXQR3fyXhjcXNnHEIsblqLrMCal3pLxs7lI+KPYMa2ZYL/am4
P"}]}

 - match: { license_status:  "valid" }

 - do:
     xpack.license.get: {}

 - length: { license: 11 }

 ## license version: 1.x
 - do:
     xpack.license.post:
       acknowledge: true
       body: |
         {"licenses":[{"uid":"893361dc-9749-4997-93cb-
802e3d7fa4a8","type":"subscription","subscription_type":"gold","issue_date_in_millis":1411948800000,"feature":"
```

```
shield","expiry_date_in_millis":1914278399999,"max_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature
":"AAAAAQAAAA0LVAywwpSH94cyXr4zAAABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVo
MVZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0pp
THBkWkprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeG
wxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEphZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNi
WktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQaVB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K
055c28zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2VE1qbVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9w
TTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldMbmMzSGtQc3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2
bDRWUzh6UG5pdENGWFRreXNKNkE9PQAAAQA4qscc/URRZVdFoLwgy9dqybYEQLW8YLkiAyPV5XHHH
dtk+dtZIepiNEDkUXhSX2waVJlsNRF8/4kqplDfwNoD2TUM8fTgiIfiSiZYGDTGST+yW/5eAveEU5J5v1liBN27b
wkqL+V4YAa0Tcm7NKKwjScWKAHiTU3vF8chPkGfCHE0kQgVwPC9RE82pTw0s6/uR4PfLGNFfqPM0uiE5n
ucfVrtj89JQiO/KA/7ZyFbo7VTNXxZQt7T7rZWBCP9KIjptXzcWuk08Q5S+rSoJNYbFo3HGKtrCVsRz/55rceNtd
wKKXu1IwnSeir4I1/KLduQTtFLy0+1th87VS8T88UT"}]}

 - match: { license_status:  "valid" }

 - do:
     xpack.license.get: {}

 - length: { license: 11 }

 ## multiple licenses version: 1.x
 - do:
     xpack.license.post:
       acknowledge: true
       body: |
         {"licenses":[{"uid":"893361dc-9749-4997-93cb-
802e3d7fa4a8","type":"internal","subscription_type":"none","issue_date_in_millis":1411948800000,"feature":"shiel
d","expiry_date_in_millis":1440892799999,"max_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"A
AAAAQAAAA04Q4ky3rFyyWLFkytEAAABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZw
UzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0ppTHBkW
kprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeGwxb0U0
MXZ0dDJHSERiZTVLOHNzSDByWnpoZEphZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNW
GM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQaVB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c28z
SmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2VE1qbVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9wTTJEND
VvT1NCYklacUYyK2FwRW9xa0t6dldMbmMzSGtQc3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUz
h6UG5pdENGWFRreXNKNkE9PQAAAQBxMvUMn4h2E4R4TQMijahTxQj4LPQO4f1M79UxX/XkDlGcH+J5p
RHx08OtTRPsFL1lED+h+PIXx307Vo+PNDsOxrWvoYZeYBkOLAO3ny9vhQga+52jYhMxIuFrT9xbcSCSNpMh
GojgOIPU2WgiopVdVcimo1+Gk8VtklPB1wPwFzfOjOnPgp/Icx3WYpfkeAUUOyWUYiFIBAe4bnz84iF+xwLKb
gYk6aHF25ECBtdb/Uruhcm9+jEFpoIEUtCouvvk9C+NJZ4OickV4xpRgaRG2x9PONH8ZN0QGhGYhJGbisoCxu
DmlLsyVxqxfMu3n/r7/jdsEJScjAlSrsLDOu6H"},{"uid":"893361dc-9749-4997-93cb-
802e3dofh7aa","type":"internal","subscription_type":"none","issue_date_in_millis":1443484800000,"feature":"watc
her","expiry_date_in_millis":1914278399999,"max_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"
AAAAAQAAAA0Sc90guRIaQEmgLvMnAAABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMV
ZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0ppTHB
kWkprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeGwxb0
U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEphZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktN
WGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQaVB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c2
```

```
8zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2VE1qbVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9wTTJEN
DVvT1NCYklacUYyK2FwRW9xa0t6dldMbmMzSGtQc3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRW
Uzh6UG5pdENGWFRreXNKNkE9PQAAAQCQ94dju0pnDZR3Uuypi0ic3aQJ+nvVqe+U8u79Dga5n1qIjcHDh7H
vIBJEkF+tnVPlo/PXV/x7BZSwVY1PVErit+6rYix1yuHEgqwxmx/VdRICjCaZM6tk0Ob4dZCPv6Ebn2Mmk89KH
C/PwiLPqF6QfwV/Pkpa8k2A3ORJmvYSDvXhe6tCs8dqc4ebrsFxqrZjwWh5CZSpzqqZBFXlngDv2N0hHhpGlueR
szD0JJ5dfEL5ZA1DDOrgO9OJVejSHyRqe1L5QRUNdXPVfS+EAG0Dd1cNdJ/sMpYCPnVjbw6iq2/YgM3cuztsX
VBY7ij4WnoP3ce7Zjs9TwHn+IqzftC6"}]}

 - match: { license_status:  "valid" }

 - do:
     xpack.license.get: {}

 - length: { license: 11 }
 - match: { license.uid: "893361dc-9749-4997-93cb-802e3dofh7aa" }
---
"Should throw 404 after license deletion":
 - do:
     xpack.license.delete: {}

 - match: { acknowledged: true }

 - do:
     xpack.license.get: {}
     catch: missing

---
"Should install a feature type license":

   # VERSION_NO_FEATURE_TYPE license version
 - do:
     xpack.license.post:
       acknowledge: true
       body: |
         {"license": {"uid":"893361dc-9749-4997-93cb-
802e3d7fa4a8","type":"gold","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1914278399999,"ma
x_nodes":1,"issued_to":"issued_to","issuer":"issuer","signature":"AAAAAgAAAA3U8+YmnvwC+CWsV/mRAA
ABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N
2MXMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0ppTHBkWkprZWZSQi9iNmRQNkw1SlpLN0lDalZC
S095MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZ
EphZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pS
OEhQaVB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c28zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3
o2VE1qbVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9wTTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldMbm
MzSGtQc3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdENGWFRreXNKNkE9PQAAAQB
e8GfzDm6T537Iuuvjetb3xK5dvg0K5NQapv+rczWcQFxgCuzbF8plkgetP1aAGZP4uRESDQPMlOCsx4d0UqqAm
9f7GbBQ3l93P+PogInPFeEH9NvOmaAQovmxVM9SE6DsDqlX4cXSO+bgWpXPTd2LmpoQc1fXd6BZ8GeuyYp
VHVKp9hVU0tAYjw6HzYOE7+zuO1oJYOxElqy66AnIfkvHrvni+flym3tE7tDTgsDRaz7W3iBhaqiSntEqabEkvH
dPHQdSR99XGaEvnHO1paK01/35iZF6OXHsF7CCj+558GRXiVxzueOe7TsGSSt8g7YjZwV9bRCyU7oB4B/nidg
```

```
I"}}

 - match: { license_status:  "valid" }

 - do:
     xpack.license.get: {}

 - length: { license: 11 }
---
"Cannot start basic":

 - do:
     catch: bad_request
     xpack.license.post:
       acknowledge: true
       body: |
         {"license":{"uid":"893361dc-9749-4997-93cb-
802e3d7fa4a8","type":"basic","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1914278399999,"ma
x_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"AAAAAgAAAA0lKPZ0a7aZquUltho/AAABmC
9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXM
vRFk4d3JTZEx3R3RRZ0pzU3lobWJKZnQvSEFva0ppTHBkWkprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095M
XRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEphZH
BEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQa
VB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c28zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2VE1q
bVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9wTTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldMbmMzSGtQ
c3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdENGWFRreXNKNkE9PQAAAQAALuQ44
S3IG6SzolcXVJ6Z4CIXORDrYQ+wdLCeey0XdujTslAOj+k+vNgo6wauc7Uswi01esHu4lb5IgpvKy7RRCbh5bj/z2
ubu2qMJqopp9BQyD7VQjVfqmG6seUMJwJ1a5Avvm9r41YPSPcrii3bKK2e1l6jK6N8ibCvnTyY/XkYGCJrBWT
SJePDbg6ErbyodrZ37x1StLbPWcNAkmweyHjDJnvYnbeZZO7A3NmubXZjW7Ttf8/YwQyE00PqMcl7fVPY3hk
KpAeHf8aaJbqkKYbqZuER3EWJX7ZvLVb1dNdNg8aXRn7YrkQcYwWgptYQpfV+D7yEJ4j5muAEoler"}}

 - match: { error.root_cause.0.reason: 'Installing basic licenses is no longer allowed. Use the POST
/_xpack/license/start_basic API to install a basic license that does not expire.' }
---
"Should fail gracefully when body content is not provided":

 - do:
     catch: bad_request
     xpack.license.post:
       acknowledge: true

 - match: { error.root_cause.0.reason: 'The license must be provided in the request body' }
---
"Current license is trial means not eligle to start trial":

 - do:
     xpack.license.get_trial_status: {}
 - match: { eligible_to_start_trial: false }

 - do:
     xpack.license.post_start_basic:
       acknowledge: true

 - match: { basic_was_started: true }

 - do:
     xpack.license.get_trial_status: {}

 - match: { eligible_to_start_trial: false }

 - do:
     catch: forbidden
     xpack.license.post_start_trial:
       acknowledge: true

 - match: { trial_was_started: false }
 - match: { error_message: "Operation failed: Trial was already activated." }
---
"Trial license cannot be basic":
 - do:
     catch: bad_request
     xpack.license.post_start_trial:
       type: "basic"
       acknowledge: true
---
"Can start basic license if do not already have basic":
 - do:
     xpack.license.get_basic_status: {}

 - match: { eligible_to_start_basic: true }

 - do:
     xpack.license.post_start_basic:
       acknowledge: true

 - match: { basic_was_started: true }
 - match: { acknowledged: true }

 - do:
       xpack.license.get_basic_status: {}

 - match: { eligible_to_start_basic: false }

 - do:
     catch: forbidden
     xpack.license.post_start_basic: {}

 - match: { basic_was_started: false }
 - match: { acknowledged: true }
 - match: { error_message: "Operation failed: Current license is basic." }
---
"Must acknowledge to start basic":
 - do:
     xpack.license.post_start_basic: {}

 - match: { basic_was_started: false }
 - match: { acknowledged: false }
 - match: { error_message: "Operation failed: Needs acknowledgement." }
{
 "xpack.license.delete": {
   "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
   "methods": ["DELETE"],
   "url": {
     "path": "/_xpack/license",
     "paths": ["/_xpack/license"],
     "parts" : {}
   },
   "body": null
 }
}
{
 "xpack.license.get": {
   "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
   "methods": ["GET"],
   "url": {
     "path": "/_xpack/license",
     "paths": ["/_xpack/license"],
     "parts" : {
     },
     "params": {
       "local": {
         "type" : "boolean",
         "description" : "Return local information, do not retrieve the state from master node (default: false)"
       }
     }
   },
   "body": null
 }
}
{
 "xpack.license.post_start_basic": {
   "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
   "methods": ["POST"],
   "url": {
     "path": "/_xpack/license/start_basic",
     "paths": ["/_xpack/license/start_basic"],
     "parts" : {
     },
     "params": {
       "acknowledge": {
         "type" : "boolean",
         "description" : "whether the user has acknowledged acknowledge messages (default: false)"
       }
     }
   },
   "body": null
 }
}
{
 "xpack.license.get_trial_status": {
   "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
   "methods": ["GET"],
   "url": {
     "path": "/_xpack/license/trial_status",
     "paths": ["/_xpack/license/trial_status"],
     "parts" : {
     },
     "params": {
     }
   },
   "body": null
 }
}
{
 "xpack.license.post": {
   "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
   "methods": ["PUT", "POST"],
   "url": {
     "path": "/_xpack/license",
     "paths": ["/_xpack/license"],
     "parts" : {
     },
     "params": {
       "acknowledge": {
         "type" : "boolean",
         "description" : "whether the user has acknowledged acknowledge messages (default: false)"
       }
     }
   },
   "body": {
     "description" : "licenses to be installed"
```

```
    }
  }
}
{
  "xpack.license.post_start_trial": {
    "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
    "methods": ["POST"],
    "url": {
      "path": "/_xpack/license/start_trial",
      "paths": ["/_xpack/license/start_trial"],
      "parts" : {
      },
      "params": {
        "type": {
          "type" : "string",
          "description" : "The type of trial license to generate (default: \"trial\")"
        },
        "acknowledge": {
          "type" : "boolean",
          "description" : "whether the user has acknowledged acknowledge messages (default: false)"
        }
      }
    },
    "body": null
  }
}
{
  "xpack.license.get_basic_status": {
    "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
    "methods": ["GET"],
    "url": {
      "path": "/_xpack/license/basic_status",
      "paths": ["/_xpack/license/basic_status"],
      "parts" : {
      },
      "params": {
      }
    },
    "body": null
  }
}
```

Copyright (c) 2004-2017 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Dropwizard
Copyright 2010-2013 Coda Hale and Yammer, Inc., 2014-2016 Dropwizard Team

This product includes software developed by Coda Hale and Yammer, Inc.

Apache Log4j SLF4J Binding
Copyright 1999-2017 The Apache Software Foundation

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

Apache XML Security for Java
Copyright 2000-2016 The Apache Software Foundation

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

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="utf-8">
 <link rel="dns-prefetch" href="https://assets-cdn.github.com">
 <link rel="dns-prefetch" href="https://avatars0.githubusercontent.com">
 <link rel="dns-prefetch" href="https://avatars1.githubusercontent.com">
 <link rel="dns-prefetch" href="https://avatars2.githubusercontent.com">
 <link rel="dns-prefetch" href="https://avatars3.githubusercontent.com">
 <link rel="dns-prefetch" href="https://github-cloud.s3.amazonaws.com">
 <link rel="dns-prefetch" href="https://user-images.githubusercontent.com/">
```

<link crossorigin="anonymous" href="https://assets-cdn.github.com/assets/frameworks-2d2d4c150f7000385741c6b992b302689ecd172246c6428904e0813be9bceca6.css" media="all" rel="stylesheet" />
<link crossorigin="anonymous" href="https://assets-cdn.github.com/assets/github-0522ae8d3b3bdc841d2f91f90efd5f1fd9040d910905674cd134ced43a6dfea6.css" media="all" rel="stylesheet" />

<link crossorigin="anonymous" href="https://assets-cdn.github.com/assets/site-cfab053e93f0e27f4c63d4ff6b7957bd25f711667fe678e747f8a4d88c47b38d.css" media="all" rel="stylesheet" />

<meta name="viewport" content="width=device-width">

<title>cryptacular/NOTICE at master vt-middleware/cryptacular GitHub</title>
<link rel="search" type="application/opensearchdescription+xml" href="/opensearch.xml" title="GitHub">
<link rel="fluid-icon" href="https://github.com/fluidicon.png" title="GitHub">
<meta property="fb:app_id" content="1401488693436528">

<meta content="https://avatars7.githubusercontent.com/u/6122907?v=4&s=400" property="og:image" /><meta content="GitHub" property="og:site_name" /><meta content="object" property="og:type" /><meta content="vt-middleware/cryptacular" property="og:title" /><meta content="https://github.com/vt-middleware/cryptacular" property="og:url" /><meta content="cryptacular - The friendly complement to the BouncyCastle crypto API for Java." property="og:description" />

<link rel="assets" href="https://assets-cdn.github.com/">

<meta name="pjax-timeout" content="1000">

<meta name="request-id" content="E0E4:26F16:12A5AE6:1D11801:596C7978" data-pjax-transient>

<meta name="selected-link" value="repo_source" data-pjax-transient>

<meta name="google-site-verification" content="KT5gs8h0wvaagLKAVWq8bbeNwnZZK1r1XQysX3xurLU">
<meta name="google-site-verification" content="ZzhVyEFwb7w3e0-uOTltm8Jsck2F5StVihD0exw2fsA">
<meta name="google-analytics" content="UA-3769691-2">

<meta content="collector.githubapp.com" name="octolytics-host" /><meta content="github" name="octolytics-app-id" /><meta content="https://collector.githubapp.com/github-external/browser_event" name="octolytics-event-url" /><meta content="E0E4:26F16:12A5AE6:1D11801:596C7978" name="octolytics-dimension-request_id" /><meta content="sea" name="octolytics-dimension-region_edge" /><meta content="iad" name="octolytics-dimension-region_render" />
<meta content="/<user-name>/<repo-name>/blob/show" data-pjax-transient="true" name="analytics-location" />

```
 <meta class="js-ga-set" name="dimension1" content="Logged Out">



    <meta name="hostname" content="github.com">
 <meta name="user-login" content="">

    <meta name="expected-hostname" content="github.com">
  <meta name="js-proxy-site-detection-payload" content="N2ZhMjk0NTA4MjI1NmZhYTVlNzM5NzVjZmFkOWY2NGFkNmMxYzcyMGViNzAzZGQxMGMzZmJhZDQ3YWZiZTI0OHx7InJlbW90ZV9hZGRyZXNzIjoiMTEwLjIwLjIyMC4xMzUiLCJyZXF1ZXN0X2lkIjoiRTBFNDoyNkYxNjoxMkE1QUU2OjFEMTE4MDE6NTk2Qzc5NzgiLCJ0aW1lc3RhbXAiOjE1MDAyODEyMDgsImhvc3QiOiJnaXRodWIuY29tIn0=">


 <meta name="html-safe-nonce" content="5aa226b80a18dc40894e1d405e4eb31cfca7d616">

 <meta http-equiv="x-pjax-version" content="f682644ce1bb9629b9d9d9bedf64801b">


    <link href="https://github.com/vt-middleware/cryptacular/commits/master.atom" rel="alternate" title="Recent Commits to cryptacular:master" type="application/atom+xml">

 <meta name="description" content="cryptacular - The friendly complement to the BouncyCastle crypto API for Java.">
 <meta name="go-import" content="github.com/vt-middleware/cryptacular git https://github.com/vt-middleware/cryptacular.git">

 <meta content="6122907" name="octolytics-dimension-user_id" /><meta content="vt-middleware" name="octolytics-dimension-user_login" /><meta content="15714989" name="octolytics-dimension-repository_id" /><meta content="vt-middleware/cryptacular" name="octolytics-dimension-repository_nwo" /><meta content="true" name="octolytics-dimension-repository_public" /><meta content="false" name="octolytics-dimension-repository_is_fork" /><meta content="15714989" name="octolytics-dimension-repository_network_root_id" /><meta content="vt-middleware/cryptacular" name="octolytics-dimension-repository_network_root_nwo" /><meta content="false" name="octolytics-dimension-repository_explore_github_marketplace_ci_cta_shown" />


  <link rel="canonical" href="https://github.com/vt-middleware/cryptacular/blob/master/NOTICE" data-pjax-transient>


 <meta name="browser-stats-url" content="https://api.github.com/_private/browser/stats">

 <meta name="browser-errors-url" content="https://api.github.com/_private/browser/errors">
 <link rel="mask-icon" href="https://assets-cdn.github.com/pinned-octocat.svg" color="#000000">
 <link rel="icon" type="image/x-icon" href="https://assets-cdn.github.com/favicon.ico">

<meta name="theme-color" content="#1e2327">




 </head>

 <body class="logged-out env-production page-blob">




 <div class="position-relative js-header-wrapper ">
   <a href="#start-of-content" tabindex="1" class="px-2 py-4 show-on-focus js-skip-to-content">Skip to content</a>
   <div id="js-pjax-loader-bar" class="pjax-loader-bar"><div class="progress"></div></div>







       <header class="site-header js-details-container Details" role="banner">
 <div class="site-nav-container">
   <a class="header-logo-invertocat" href="https://github.com/" aria-label="Homepage" data-ga-click="(Logged out)
Header, go to homepage, icon:logo-wordmark">
     <svg aria-hidden="true" class="octicon octicon-mark-github" height="32" version="1.1" viewBox="0 0 16 16"
width="32"><path fill-rule="evenodd" d="M8 0C3.58 0 0 3.58 0 8c0 3.54 2.29 6.53 5.47 7.59.4.07.55-.17.55-.38 0-
.19-.01-.82-.01-1.49-2.01.37-2.53-.49-2.69-.94-.09-.23-.48-.94-.82-1.13-.28-.15-.68-.52-.01-.53.63-.01 1.08.58
1.23.82.72 1.21 1.87.87 2.33.66.07-.52.28-.87.51-1.07-1.78-.2-3.64-.89-3.64-3.95 0-.87.31-1.59.82-2.15-.08-.2-.36-
1.02.08-2.12 0 0 .67-.21 2.2.82.64-.18 1.32-.27 2-.27.68 0 1.36.09 2 .27 1.53-1.04 2.2-.82 2.2-.82.44 1.1.16 1.92.08
2.12.51.56.82 1.27.82 2.15 0 3.07-1.87 3.75-3.65 3.95.29.25.54.73.54 1.48 0 1.07-.01 1.93-.01 2.2 0
.21.15.46.55.38A8.013 8.013 0 0 0 16 8c0-4.42-3.58-8-8-8z"/></svg>
   </a>

   <button class="btn-link float-right site-header-toggle js-details-target" type="button" aria-label="Toggle
navigation" aria-expanded="false">
     <svg aria-hidden="true" class="octicon octicon-three-bars" height="24" version="1.1" viewBox="0 0 12 16"
width="18"><path fill-rule="evenodd" d="M11.41 9H.59C0 9 0 8.59 0 8c0-.59 0-1 .59-1H11.4c.59 0 .59.41.59 1 0
.59 0 1-.59 1h.01zm0-4H.59C0 5 0 4.59 0 4c0-.59 0-1 .59-1H11.4c.59 0 .59.41.59 1 0 .59 0 1-.59 1h.01zM.59
11H11.4c.59 0 .59.41.59 1 0 .59 0 1-.59 1H.59C0 13 0 12.59 0 12c0-.59 0-1 .59-1z"/></svg>
   </button>

   <div class="site-header-menu">
     <nav class="site-header-nav">
       <a href="/features" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav menu -
```

```
item:features" data-selected-links="/features /features/code-review /features/project-management
/features/integrations /features">
        Features
</a>        <a href="/business" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav menu
- item:business" data-selected-links="/business /business/security /business/customers /business">
        Business
</a>        <a href="/explore" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav menu -
item:explore" data-selected-links="/explore /trending /trending/developers /stars /integrations
/integrations/feature/code /integrations/feature/collaborate /integrations/feature/ship showcases showcases_search
showcases_landing /explore">
        Explore
</a>            <a href="/marketplace" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav
menu - item:marketplace" data-selected-links=" /marketplace">
            Marketplace
</a>        <a href="/pricing" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav menu -
item:pricing" data-selected-links="/pricing /pricing/developer /pricing/team /pricing/business-hosted
/pricing/business-enterprise /pricing">
        Pricing
</a>      </nav>

     <div class="site-header-actions">
        <div class="header-search scoped-search site-scoped-search js-site-search" role="search">
 <!-- '"` --><!-- </textarea></xmp> --></option></form><form accept-charset="UTF-8" action="/vt-
middleware/cryptacular/search" class="js-site-search-form" data-scoped-search-url="/vt-
middleware/cryptacular/search" data-unscoped-search-url="/search" method="get"><div
style="margin:0;padding:0;display:inline"><input name="utf8" type="hidden" value="&#x2713;" /></div>
   <label class="form-control header-search-wrapper js-chromeless-input-container">
      <a href="/vt-middleware/cryptacular/blob/master/NOTICE" class="header-search-scope no-underline">This
repository</a>
     <input type="text"
      class="form-control header-search-input js-site-search-focus js-site-search-field is-clearable"
      data-hotkey="s"
      name="q"
      value=""
      placeholder="Search"
      aria-label="Search this repository"
      data-unscoped-placeholder="Search GitHub"
      data-scoped-placeholder="Search"
      autocapitalize="off">
      <input type="hidden" class="js-site-search-type-field" name="type" >
   </label>
</form></div>



        <a class="text-bold site-header-link" href="/login?return_to=%2Fvt-
middleware%2Fcryptacular%2Fblob%2Fmaster%2FNOTICE" data-ga-click="(Logged out) Header, clicked Sign
in, text:sign-in">Sign in</a>
          <span class="text-gray">or</span>
          <a class="text-bold site-header-link" href="/join?source=header-repo" data-ga-click="(Logged out) Header, clicked Sign up, text:sign-up">Sign up</a>
     </div>
   </div>
 </div>
</header>


 </div>

 <div id="start-of-content" class="show-on-focus"></div>

   <div id="js-flash-container">
</div>



 <div role="main">
       <div itemscope itemtype="http://schema.org/SoftwareSourceCode">
   <div id="js-repo-pjax-container" data-pjax-container>







   <div class="pagehead repohead instapaper_ignore readability-menu experiment-repo-nav">
     <div class="container repohead-details-container">

       <ul class="pagehead-actions">
 <li>
     <a href="/login?return_to=%2Fvt-middleware%2Fcryptacular"
   class="btn btn-sm btn-with-count tooltipped tooltipped-n"
   aria-label="You must be signed in to watch a repository" rel="nofollow">
   <svg aria-hidden="true" class="octicon octicon-eye" height="16" version="1.1" viewBox="0 0 16 16" width="16"><path fill-rule="evenodd" d="M8.06 2C3 2 0 8 0 8s3 6 8.06 6C13 14 16 8 16 8s-3-6-7.94-6zM8 12c-2.2 0-4-1.78-4-4 0-2.2 1.8-4 4-4 2.22 0 4 1.8 4 4 0 2.22-1.78 4-4 4zm2-4c0 1.11-.89 2-2 2-1.11 0-2-.89-2-2 0-1.11.89-2 2-2 1.11 0 2 .89 2 2z"/></svg>
   Watch
 </a>
 <a class="social-count" href="/vt-middleware/cryptacular/watchers"
    aria-label="9 users are watching this repository">
   9
 </a>

 </li>
 <li>
     <a href="/login?return_to=%2Fvt-middleware%2Fcryptacular"
   class="btn btn-sm btn-with-count tooltipped tooltipped-n"
   aria-label="You must be signed in to star a repository" rel="nofollow">
   <svg aria-hidden="true" class="octicon octicon-star" height="16" version="1.1" viewBox="0 0 14 16" width="14"><path fill-rule="evenodd" d="M14 6l-4.9-.64L7 1 4.9 5.36 0 6l3.6 3.26L2.67 14 7 11.67 11.33 14l-.93-4.74z"/></svg>
   Star
 </a>

   <a class="social-count js-social-count" href="/vt-middleware/cryptacular/stargazers"
     aria-label="15 users starred this repository">
     15
   </a>

 </li>

 <li>
     <a href="/login?return_to=%2Fvt-middleware%2Fcryptacular"
       class="btn btn-sm btn-with-count tooltipped tooltipped-n"
       aria-label="You must be signed in to fork a repository" rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-repo-forked" height="16" version="1.1" viewBox="0 0 10 16" width="10"><path fill-rule="evenodd" d="M8 1a1.993 1.993 0 0 0-1 3.72V6L5 8 3 6V4.72A1.993 1.993 0 0 0 2 1a1.993 1.993 0 0 0-1 3.72V6.5l3 3v1.78A1.993 1.993 0 0 0 5 15a1.993 1.993 0 0 0 1-3.72V9.5l3-3V4.72A1.993 1.993 0 0 0 8 1zM2 4.2C1.34 4.2.8 3.65.8 3c0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2zm3 10c-.66 0-1.2-.55-1.2-1.2 0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2zm3-10c-.66 0-1.2-.55-1.2-1.2 0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2z"/></svg>
       Fork
     </a>

   <a href="/vt-middleware/cryptacular/network" class="social-count"
      aria-label="1 user forked this repository">
     1
   </a>
 </li>
</ul>

       <h1 class="public ">
 <svg aria-hidden="true" class="octicon octicon-repo" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M4 9H3V8h1v1zm0-3H3v1h1V6zm0-2H3v1h1V4zm0-2H3v1h1V2zm8-1v12c0 .55-.45 1-1 1H6v2l-1.5-1.5L3 16v-2H1c-.55 0-1-.45-1-1V1c0-.55.45-1 1-1h10c.55 0 1 .45 1 1zm-1 10H1v2h2v-1h3v1h5v-2zm0-10H2v9h9V1z"/></svg>
 <span class="author" itemprop="author"><a href="/vt-middleware" class="url fn" rel="author">vt-middleware</a></span><!--
--><span class="path-divider">/</span><!--
--><strong itemprop="name"><a href="/vt-middleware/cryptacular" data-pjax="#js-repo-pjax-container">cryptacular</a></strong>
```

```
</h1>

     </div>
     <div class="container">

<nav class="reponav js-repo-nav js-sidenav-container-pjax"
    itemscope
    itemtype="http://schema.org/BreadcrumbList"
    role="navigation"
    data-pjax="#js-repo-pjax-container">

 <span itemscope itemtype="http://schema.org/ListItem" itemprop="itemListElement">
   <a href="/vt-middleware/cryptacular" class="js-selected-navigation-item selected reponav-item" data-hotkey="g c" data-selected-links="repo_source repo_downloads repo_commits repo_releases repo_tags repo_branches /vt-middleware/cryptacular" itemprop="url">
     <svg aria-hidden="true" class="octicon octicon-code" height="16" version="1.1" viewBox="0 0 14 16" width="14"><path fill-rule="evenodd" d="M9.5 3L8 4.5 11.5 8 8 11.5 9.5 13 14 8 9.5 3zm-5 0L0 8l4.5 5L6 11.5 2.5 8 6 4.5 4.5 3z"/></svg>
     <span itemprop="name">Code</span>
     <meta itemprop="position" content="1">
</a>  </span>

   <span itemscope itemtype="http://schema.org/ListItem" itemprop="itemListElement">
     <a href="/vt-middleware/cryptacular/issues" class="js-selected-navigation-item reponav-item" data-hotkey="g i" data-selected-links="repo_issues repo_labels repo_milestones /vt-middleware/cryptacular/issues" itemprop="url">
       <svg aria-hidden="true" class="octicon octicon-issue-opened" height="16" version="1.1" viewBox="0 0 14 16" width="14"><path fill-rule="evenodd" d="M7 2.3c3.14 0 5.7 2.56 5.7 5.7s-2.56 5.7-5.7 5.7A5.71 5.71 0 0 1 1.3 8c0-3.14 2.56-5.7 5.7-5.7zM7 1C3.14 1 0 4.14 0 8s3.14 7 7 7 7-3.14 7-7-3.14-7-7-7zm1 3H6v5h2V4zm0 6H6v2h2v-2z"/></svg>
       <span itemprop="name">Issues</span>
       <span class="Counter">4</span>
       <meta itemprop="position" content="2">
</a>    </span>

 <span itemscope itemtype="http://schema.org/ListItem" itemprop="itemListElement">
   <a href="/vt-middleware/cryptacular/pulls" class="js-selected-navigation-item reponav-item" data-hotkey="g p" data-selected-links="repo_pulls /vt-middleware/cryptacular/pulls" itemprop="url">
     <svg aria-hidden="true" class="octicon octicon-git-pull-request" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M11 11.28V5c-.03-.78-.34-1.47-.94-2.06C9.46 2.35 8.78 2.03 8 2H7V0L4 3l3 3V4h1c.27.02.48.11.69.31.21.2.3.42.31.69v6.28A1.993 1.993 0 0 0 10 15a1.993 1.993 0 0 0 1-3.72zm-1 2.92c-.66 0-1.2-.55-1.2-1.2 0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2zM4 3c0-1.11-.89-2-2-2a1.993 1.993 0 0 0-1 3.72v6.56A1.993 1.993 0 0 0 2 15a1.993 1.993 0 0 0 1-3.72V4.72c.59-.34 1-.98 1-1.72zm-.8 10c0 .66-.55 1.2-1.2 1.2-.65 0-1.2-.55-1.2-1.2 0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2zM2 4.2C1.34 4.2.8 3.65.8 3c0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2z"/></svg>
     <span itemprop="name">Pull requests</span>
     <span class="Counter">1</span>
     <meta itemprop="position" content="3">
</a>  </span>
```

```
  <a href="/vt-middleware/cryptacular/projects" class="js-selected-navigation-item reponav-item" data-selected-links="repo_projects new_repo_project repo_project /vt-middleware/cryptacular/projects">
    <svg aria-hidden="true" class="octicon octicon-project" height="16" version="1.1" viewBox="0 0 15 16" width="15"><path fill-rule="evenodd" d="M10 12h3V2h-3v10zm-4-2h3V2H6v8zm-4 4h3V2H2v12zm-1 1h13V1H1v14zM14 0H1a1 1 0 0 0-1 1v14a1 1 0 0 0 1 1h13a1 1 0 0 0 1-1V1a1 1 0 0 0-1-1z"/></svg>
    Projects
    <span class="Counter" >0</span>
</a>


  <div class="reponav-dropdown js-menu-container">
    <button type="button" class="btn-link reponav-item reponav-dropdown js-menu-target " data-no-toggle aria-expanded="false" aria-haspopup="true">
      Insights
      <svg aria-hidden="true" class="octicon octicon-triangle-down v-align-middle text-gray" height="11" version="1.1" viewBox="0 0 12 16" width="8"><path fill-rule="evenodd" d="M0 5l6 6 6-6z"/></svg>
    </button>
    <div class="dropdown-menu-content js-menu-content">
      <div class="dropdown-menu dropdown-menu-sw">
        <a class="dropdown-item" href="/vt-middleware/cryptacular/pulse" data-skip-pjax>
          <svg aria-hidden="true" class="octicon octicon-pulse" height="16" version="1.1" viewBox="0 0 14 16" width="14"><path fill-rule="evenodd" d="M11.5 8L8.8 5.4 6.6 8.5 5.5 1.6 2.38 8H0v2h3.6l.9-1.8.9 5.4L9 8.5l1.6 1.5H14V8z"/></svg>
          Pulse
        </a>
        <a class="dropdown-item" href="/vt-middleware/cryptacular/graphs" data-skip-pjax>
          <svg aria-hidden="true" class="octicon octicon-graph" height="16" version="1.1" viewBox="0 0 16 16" width="16"><path fill-rule="evenodd" d="M16 14v1H0V0h1v14h15zM5 13H3V8h2v5zm4 0H7V3h2v10zm4 0h-2V6h2v7z"/></svg>
          Graphs
        </a>
      </div>
    </div>
  </div>
</nav>

    </div>
  </div>

<div class="container new-discussion-timeline experiment-repo-nav">
 <div class="repository-content">



 <a href="/vt-middleware/cryptacular/blob/c26911e3cd28497ce9daa3ce682e09cb2d1d8688/NOTICE" class="d-none js-permalink-shortcut" data-hotkey="y">Permalink</a>

 <!-- blob contrib key: blob_contributors:v21:f5054b9e46039e0ad937c69e6151b7d4 -->
 <div class="file-navigation js-zeroclipboard-container">

<div class="select-menu branch-select-menu js-menu-container js-select-menu float-left">
 <button class=" btn btn-sm select-menu-button js-menu-target css-truncate" data-hotkey="w"

   type="button" aria-label="Switch branches or tags" aria-expanded="false" aria-haspopup="true">
     <i>Branch:</i>
     <span class="js-select-button css-truncate-target">master</span>
 </button>

 <div class="select-menu-modal-holder js-menu-content js-navigation-container" data-pjax>

   <div class="select-menu-modal">
     <div class="select-menu-header">
       <svg aria-label="Close" class="octicon octicon-x js-menu-close" height="16" role="img" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M7.48 8l3.75 3.75-1.48 1.48L6 9.48l-3.75 3.75-
1.48-1.48L4.52 8 .77 4.25l1.48-1.48L6 6.52l3.75-3.75 1.48 1.48z"/></svg>
       <span class="select-menu-title">Switch branches/tags</span>
     </div>

     <div class="select-menu-filters">
       <div class="select-menu-text-filter">
         <input type="text" aria-label="Filter branches/tags" id="context-commitish-filter-field" class="form-control
js-filterable-field js-navigation-enable" placeholder="Filter branches/tags">
       </div>
       <div class="select-menu-tabs">
         <ul>
           <li class="select-menu-tab">
             <a href="#" data-tab-filter="branches" data-filter-placeholder="Filter branches/tags" class="js-select-menu-
tab" role="tab">Branches</a>
           </li>
           <li class="select-menu-tab">
             <a href="#" data-tab-filter="tags" data-filter-placeholder="Find a tag" class="js-select-menu-tab"
role="tab">Tags</a>
           </li>
         </ul>
       </div>
     </div>

     <div class="select-menu-list select-menu-tab-bucket js-select-menu-tab-bucket" data-tab-filter="branches"
role="menu">

       <div data-filterable-for="context-commitish-filter-field" data-filterable-type="substring">



           <a class="select-menu-item js-navigation-item js-navigation-open "
              href="/vt-middleware/cryptacular/blob/ISSUE-31+32/NOTICE"
```

```
        data-name="ISSUE-31+32"
        data-skip-pjax="true"
        rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target js-select-menu-filter-text">
        ISSUE-31+32
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
        href="/vt-middleware/cryptacular/blob/gh-pages/NOTICE"
        data-name="gh-pages"
        data-skip-pjax="true"
        rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target js-select-menu-filter-text">
        gh-pages
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open selected"
        href="/vt-middleware/cryptacular/blob/master/NOTICE"
        data-name="master"
        data-skip-pjax="true"
        rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target js-select-menu-filter-text">
        master
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
        href="/vt-middleware/cryptacular/blob/v1.1/NOTICE"
        data-name="v1.1"
        data-skip-pjax="true"
        rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target js-select-menu-filter-text">
        v1.1
       </span>
      </a>
    </div>

     <div class="select-menu-no-results">Nothing to show</div>
   </div>

   <div class="select-menu-list select-menu-tab-bucket js-select-menu-tab-bucket" data-tab-filter="tags">
```

```
    <div data-filterable-for="context-commitish-filter-field" data-filterable-type="substring">


      <a class="select-menu-item js-navigation-item js-navigation-open "
        href="/vt-middleware/cryptacular/tree/v1.2.1/NOTICE"
        data-name="v1.2.1"
        data-skip-pjax="true"
        rel="nofollow">
        <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
        <span class="select-menu-item-text css-truncate-target" title="v1.2.1">
          v1.2.1
        </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
        href="/vt-middleware/cryptacular/tree/v1.2.0/NOTICE"
        data-name="v1.2.0"
        data-skip-pjax="true"
        rel="nofollow">
        <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
        <span class="select-menu-item-text css-truncate-target" title="v1.2.0">
          v1.2.0
        </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
        href="/vt-middleware/cryptacular/tree/v1.1.2/NOTICE"
        data-name="v1.1.2"
        data-skip-pjax="true"
        rel="nofollow">
        <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
        <span class="select-menu-item-text css-truncate-target" title="v1.1.2">
          v1.1.2
        </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
        href="/vt-middleware/cryptacular/tree/v1.1.1/NOTICE"
        data-name="v1.1.1"
        data-skip-pjax="true"
        rel="nofollow">
        <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
        <span class="select-menu-item-text css-truncate-target" title="v1.1.1">
          v1.1.1
        </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
```

```
       href="/vt-middleware/cryptacular/tree/v1.1.0/NOTICE"
       data-name="v1.1.0"
       data-skip-pjax="true"
       rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target" title="v1.1.0">
        v1.1.0
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
       href="/vt-middleware/cryptacular/tree/v1.0/NOTICE"
       data-name="v1.0"
       data-skip-pjax="true"
       rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target" title="v1.0">
        v1.0
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
       href="/vt-middleware/cryptacular/tree/v1.0-RC6/NOTICE"
       data-name="v1.0-RC6"
       data-skip-pjax="true"
       rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target" title="v1.0-RC6">
        v1.0-RC6
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
       href="/vt-middleware/cryptacular/tree/v1.0-RC4/NOTICE"
       data-name="v1.0-RC4"
       data-skip-pjax="true"
       rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target" title="v1.0-RC4">
        v1.0-RC4
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
       href="/vt-middleware/cryptacular/tree/v1.0-RC3/NOTICE"
       data-name="v1.0-RC3"
       data-skip-pjax="true"
       rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target" title="v1.0-RC3">
        v1.0-RC3
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
       href="/vt-middleware/cryptacular/tree/v1.0-RC2/NOTICE"
       data-name="v1.0-RC2"
       data-skip-pjax="true"
       rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target" title="v1.0-RC2">
        v1.0-RC2
       </span>
      </a>
      <a class="select-menu-item js-navigation-item js-navigation-open "
       href="/vt-middleware/cryptacular/tree/v1.0-RC1/NOTICE"
       data-name="v1.0-RC1"
       data-skip-pjax="true"
       rel="nofollow">
       <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
       <span class="select-menu-item-text css-truncate-target" title="v1.0-RC1">
        v1.0-RC1
       </span>
      </a>
    </div>

    <div class="select-menu-no-results">Nothing to show</div>
   </div>

  </div>
 </div>
</div>

  <div class="BtnGroup float-right">
   <a href="/vt-middleware/cryptacular/find/master"
      class="js-pjax-capture-input btn btn-sm BtnGroup-item"
      data-pjax
      data-hotkey="t">
    Find file
   </a>
   <button aria-label="Copy file path to clipboard" class="js-zeroclipboard btn btn-sm BtnGroup-item tooltipped tooltipped-s" data-copied-hint="Copied!" type="button">Copy path</button>
  </div>
  <div class="breadcrumb js-zeroclipboard-target">
```

```
     <span class="repo-root js-repo-root"><span class="js-path-segment"><a href="/vt-
middleware/cryptacular"><span>cryptacular</span></a></span></span><span class="separator">/</span><strong
class="final-path">NOTICE</strong>
   </div>
 </div>




 <div class="commit-tease">
     <span class="float-right">
       <a class="commit-tease-sha" href="/vt-
middleware/cryptacular/commit/6dd6f199ac3ecc3b4c5aef9e04be3bbe265a30a1" data-pjax>
         6dd6f19
       </a>
       <relative-time datetime="2017-07-06T22:28:36Z">Jul 7, 2017</relative-time>
     </span>
     <div>
       <img alt="@dfish3r" class="avatar" height="20"
src="https://avatars6.githubusercontent.com/u/1051499?v=4&s=40" width="20" />
       <a href="/dfish3r" class="user-mention" rel="contributor">dfish3r</a>
         <a href="/vt-middleware/cryptacular/commit/6dd6f199ac3ecc3b4c5aef9e04be3bbe265a30a1"
class="message" data-pjax="true" title="Update year in notice.">Update year in notice.</a>
     </div>

   <div class="commit-tease-contributors">
     <button type="button" class="btn-link muted-link contributors-toggle" data-facebox="#blob_contributors_box">
       <strong>1</strong>
        contributor
     </button>

   </div>

   <div id="blob_contributors_box" style="display:none">
     <h2 class="facebox-header" data-facebox-id="facebox-header">Users who have contributed to this file</h2>
     <ul class="facebox-user-list" data-facebox-id="facebox-description">
         <li class="facebox-user-list-item">
           <img alt="@dfish3r" height="24" src="https://avatars4.githubusercontent.com/u/1051499?v=4&s=48"
width="24" />
           <a href="/dfish3r">dfish3r</a>
         </li>
     </ul>
   </div>
 </div>

 <div class="file">
   <div class="file-header">
 <div class="file-actions">
```

```
   <div class="BtnGroup">
     <a href="/vt-middleware/cryptacular/raw/master/NOTICE" class="btn btn-sm BtnGroup-item" id="raw-url">Raw</a>
       <a href="/vt-middleware/cryptacular/blame/master/NOTICE" class="btn btn-sm js-update-url-with-hash BtnGroup-item" data-hotkey="b">Blame</a>
     <a href="/vt-middleware/cryptacular/commits/master/NOTICE" class="btn btn-sm BtnGroup-item" rel="nofollow">History</a>
   </div>



       <button type="button" class="btn-octicon disabled tooltipped tooltipped-nw"
         aria-label="You must be signed in to make or propose changes">
         <svg aria-hidden="true" class="octicon octicon-pencil" height="16" version="1.1" viewBox="0 0 14 16" width="14"><path fill-rule="evenodd" d="M0 12v3h3l8-8-3-3-8 8zm3 2H1v-2h1v1h1v1zm10.3-9.3L12 6 9 3l1.3-1.3a.996.996 0 0 1 1.41 0l1.59 1.59c.39.39.39 1.02 0 1.41z"/></svg>
       </button>
       <button type="button" class="btn-octicon btn-octicon-danger disabled tooltipped tooltipped-nw"
         aria-label="You must be signed in to make or propose changes">
         <svg aria-hidden="true" class="octicon octicon-trashcan" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M11 2H9c0-.55-.45-1-1-1H5c-.55 0-1 .45-1 1H2c-.55 0-1 .45-1 1v1c0 .55.45 1 1 1v9c0 .55.45 1 1 1h7c.55 0 1-.45 1-1V5c.55 0 1-.45 1-1V3c0-.55-.45-1-1-1zm-1 12H3V5h1v8h1V5h1v8h1V5h1v8h1V5h1v9zm1-10H2V3h9v1z"/></svg>
       </button>
  </div>

  <div class="file-info">
     7 lines (5 sloc)
     <span class="file-info-divider"></span>
   165 Bytes
  </div>
</div>




  <div itemprop="text" class="blob-wrapper data type-text">
     <table class="highlight tab-size js-file-line-container" data-tab-size="8">
     <tr>
       <td id="L1" class="blob-num js-line-number" data-line-number="1"></td>
       <td id="LC1" class="blob-code blob-code-inner js-file-line">Cryptacular Java Library</td>
     </tr>
     <tr>
       <td id="L2" class="blob-num js-line-number" data-line-number="2"></td>
       <td id="LC2" class="blob-code blob-code-inner js-file-line">Copyright (C) 2003-2017 Virginia Tech.</td>
     </tr>
     <tr>
       <td id="L3" class="blob-num js-line-number" data-line-number="3"></td>
       <td id="LC3" class="blob-code blob-code-inner js-file-line">All rights reserved.</td>
     </tr>
     <tr>
       <td id="L4" class="blob-num js-line-number" data-line-number="4"></td>
       <td id="LC4" class="blob-code blob-code-inner js-file-line">
</td>
     </tr>
     <tr>
       <td id="L5" class="blob-num js-line-number" data-line-number="5"></td>
       <td id="LC5" class="blob-code blob-code-inner js-file-line">This product includes software developed at</td>
     </tr>
     <tr>
       <td id="L6" class="blob-num js-line-number" data-line-number="6"></td>
       <td id="LC6" class="blob-code blob-code-inner js-file-line">Virginia Tech (http://www.vt.edu).</td>
     </tr>
</table>


 </div>

 </div>

 <button type="button" data-facebox="#jump-to-line" data-facebox-class="linejump" data-hotkey="l" class="d-
none">Jump to Line</button>
 <div id="jump-to-line" style="display:none">
   <!-- '"` --><!-- </textarea></xmp> --></option></form><form accept-charset="UTF-8" action="" class="js-jump-
to-line-form" method="get"><div style="margin:0;padding:0;display:inline"><input name="utf8" type="hidden"
value="&#x2713;" /></div>
     <input class="form-control linejump-input js-jump-to-line-field" type="text" placeholder="Jump to line&hellip;"
aria-label="Jump to line" autofocus>
     <button type="submit" class="btn">Go</button>
</form>  </div>

 </div>
 <div class="modal-backdrop js-touch-events"></div>
</div>

   </div>
 </div>

 </div>


<div class="container site-footer-container">
 <div class="site-footer " role="contentinfo">
   <ul class="site-footer-links float-right">
       <li><a href="https://github.com/contact" data-ga-click="Footer, go to contact, text:contact">Contact
GitHub</a></li>
     <li><a href="https://developer.github.com" data-ga-click="Footer, go to api, text:api">API</a></li>
     <li><a href="https://training.github.com" data-ga-click="Footer, go to training, text:training">Training</a></li>
```

```
    <li><a href="https://shop.github.com" data-ga-click="Footer, go to shop, text:shop">Shop</a></li>
      <li><a href="https://github.com/blog" data-ga-click="Footer, go to blog, text:blog">Blog</a></li>
      <li><a href="https://github.com/about" data-ga-click="Footer, go to about, text:about">About</a></li>

  </ul>

  <a href="https://github.com" aria-label="Homepage" class="site-footer-mark" title="GitHub">
    <svg aria-hidden="true" class="octicon octicon-mark-github" height="24" version="1.1" viewBox="0 0 16 16"
width="24"><path fill-rule="evenodd" d="M8 0C3.58 0 0 3.58 0 8c0 3.54 2.29 6.53 5.47 7.59.4.07.55-.17.55-.38 0-
.19-.01-.82-.01-1.49-2.01.37-2.53-.49-2.69-.94-.09-.23-.48-.94-.82-1.13-.28-.15-.68-.52-.01-.53.63-.01 1.08.58
1.23.82.72 1.21 1.87.87 2.33.66.07-.52.28-.87.51-1.07-1.78-.2-3.64-.89-3.64-3.95 0-.87.31-1.59.82-2.15-.08-.2-.36-
1.02.08-2.12 0 0 .67-.21 2.2.82.64-.18 1.32-.27 2-.27.68 0 1.36.09 2 .27 1.53-1.04 2.2-.82 2.2-.82.44 1.1.16 1.92.08
2.12.51.56.82 1.27.82 2.15 0 3.07-1.87 3.75-3.65 3.95.29.25.54.73.54 1.48 0 1.07-.01 1.93-.01 2.2 0
.21.15.46.55.38A8.013 8.013 0 0 0 16 8c0-4.42-3.58-8-8-8z"/></svg>
</a>
  <ul class="site-footer-links">
    <li>&copy; 2017 <span title="0.10938s from unicorn-2925809464-f2ltq">GitHub</span>, Inc.</li>
      <li><a href="https://github.com/site/terms" data-ga-click="Footer, go to terms, text:terms">Terms</a></li>
      <li><a href="https://github.com/site/privacy" data-ga-click="Footer, go to privacy,
text:privacy">Privacy</a></li>
      <li><a href="https://github.com/security" data-ga-click="Footer, go to security,
text:security">Security</a></li>
      <li><a href="https://status.github.com/" data-ga-click="Footer, go to status, text:status">Status</a></li>
      <li><a href="https://help.github.com" data-ga-click="Footer, go to help, text:help">Help</a></li>
  </ul>
 </div>
</div>



 <div id="ajax-error-message" class="ajax-error-message flash flash-error">
  <svg aria-hidden="true" class="octicon octicon-alert" height="16" version="1.1" viewBox="0 0 16 16"
width="16"><path fill-rule="evenodd" d="M8.865 1.52c-.18-.31-.51-.5-.87-.5s-.69.19-.87.5L.275 13.5c-.18.31-
.18.69 0 1 .19.31.52.5.87.5h13.7c.36 0 .69-.19.86-.5.17-.31.18-.69.01-1L8.865 1.52zM8.995 13h-2v-2h2v2zm0-3h-
2V6h2v4z"/></svg>
  <button type="button" class="flash-close js-flash-close js-ajax-error-dismiss" aria-label="Dismiss error">
    <svg aria-hidden="true" class="octicon octicon-x" height="16" version="1.1" viewBox="0 0 12 16"
width="12"><path fill-rule="evenodd" d="M7.48 8l3.75 3.75-1.48 1.48L6 9.48l-3.75 3.75-1.48-1.48L4.52 8 .77
4.25l1.48-1.48L6 6.52l3.75-3.75 1.48 1.48z"/></svg>
  </button>
  You can't perform that action at this time.
 </div>


  <script crossorigin="anonymous" src="https://assets-cdn.github.com/assets/compat-
91f98c37fc84eac24836eec2567e9912742094369a04c4eba6e3cd1fa18902d9.js"></script>
  <script crossorigin="anonymous" src="https://assets-cdn.github.com/assets/frameworks-
f84bb87b149685d1e6c6f057ee324f2cd496e677f5a359a8b5db853313bb83e6.js"></script>
```

```
   <script async="async" crossorigin="anonymous" src="https://assets-cdn.github.com/assets/github-13fa3aa50ac8f9fa9a7d198f0cd13b0905775d39446ad076d17d8f74a998438a.js"></script>




 <div class="js-stale-session-flash stale-session-flash flash flash-warn flash-banner d-none">
   <svg aria-hidden="true" class="octicon octicon-alert" height="16" version="1.1" viewBox="0 0 16 16" width="16"><path fill-rule="evenodd" d="M8.865 1.52c-.18-.31-.51-.5-.87-.5s-.69.19-.87.5L.275 13.5c-.18.31-.18.69 0 1 .19.31.52.5.87.5h13.7c.36 0 .69-.19.86-.5.17-.31.18-.69.01-1L8.865 1.52zM8.995 13h-2v-2h2v2zm0-3h-2V6h2v4z"/></svg>
   <span class="signed-in-tab-flash">You signed in with another tab or window. <a href="">Reload</a> to refresh your session.</span>
   <span class="signed-out-tab-flash">You signed out in another tab or window. <a href="">Reload</a> to refresh your session.</span>
 </div>
 <div class="facebox" id="facebox" style="display:none;">
 <div class="facebox-popup">
   <div class="facebox-content" role="dialog" aria-labelledby="facebox-header" aria-describedby="facebox-description">
   </div>
   <button type="button" class="facebox-close js-facebox-close" aria-label="Close modal">
     <svg aria-hidden="true" class="octicon octicon-x" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M7.48 8l3.75 3.75-1.48 1.48L6 9.48l-3.75 3.75-1.48-1.48L4.52 8 .77 4.25l1.48-1.48L6 6.52l3.75-3.75 1.48 1.48z"/></svg>
   </button>
 </div>
</div>



 </body>
</html>
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.security.authz.permission;

import org.elasticsearch.action.get.GetAction;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.xpack.core.security.authz.permission.Role;
import org.elasticsearch.xpack.core.security.authz.privilege.Privilege;
import org.junit.Before;

import java.util.function.Predicate;
import static org.elasticsearch.xpack.core.security.authz.privilege.IndexPrivilege.MONITOR;
import static org.elasticsearch.xpack.core.security.authz.privilege.IndexPrivilege.READ;
import static org.hamcrest.Matchers.is;
import static org.hamcrest.Matchers.notNullValue;

public class PermissionTests extends ESTestCase {
  private Role permission;

  @Before
  public void init() {
     Role.Builder builder = Role.builder("test");
     builder.add(MONITOR, "test_*", "/foo.*/");
     builder.add(READ, "baz_*foo", "/fool.*bar/");
     builder.add(MONITOR, "/bar.*/");
     permission = builder.build();
  }

  public void testAllowedIndicesMatcherAction() throws Exception {
     testAllowedIndicesMatcher(permission.indices().allowedIndicesMatcher(GetAction.NAME));
  }

  public void testAllowedIndicesMatcherActionCaching() throws Exception {
     Predicate<String> matcher1 = permission.indices().allowedIndicesMatcher(GetAction.NAME);
     Predicate<String> matcher2 = permission.indices().allowedIndicesMatcher(GetAction.NAME);
     assertThat(matcher1, is(matcher2));
  }

  public void testBuildEmptyRole() {
     Role.Builder permission = Role.builder(new String[] { "some_role" });
     Role role = permission.build();
     assertThat(role, notNullValue());
     assertThat(role.cluster(), notNullValue());
     assertThat(role.indices(), notNullValue());
     assertThat(role.runAs(), notNullValue());
  }

  public void testRunAs() {
     Role permission = Role.builder("some_role")
          .runAs(new Privilege("name", "user1", "run*"))
          .build();
     assertThat(permission.runAs().check("user1"), is(true));
     assertThat(permission.runAs().check("user"), is(false));
     assertThat(permission.runAs().check("run" + randomAlphaOfLengthBetween(1, 10)), is(true));
  }

  // "baz_*foo", "/fool.*bar/"
  private void testAllowedIndicesMatcher(Predicate<String> indicesMatcher) {
     assertThat(indicesMatcher.test("foobar"), is(false));
```

```
        assertThat(indicesMatcher.test("fool"), is(false));
        assertThat(indicesMatcher.test("fool2bar"), is(true));
        assertThat(indicesMatcher.test("baz_foo"), is(true));
        assertThat(indicesMatcher.test("barbapapa"), is(false));
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.security.authz.permission;

import org.elasticsearch.ElasticsearchParseException;
import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.xpack.core.security.authz.RoleDescriptor;
import org.elasticsearch.xpack.core.security.authz.permission.FieldPermissions;
import org.elasticsearch.xpack.core.security.authz.permission.FieldPermissionsDefinition;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.atomic.AtomicReferenceArray;

import static org.hamcrest.Matchers.containsString;

public class FieldPermissionsTests extends ESTestCase {

    public void testParseFieldPermissions() throws Exception {
        String q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
                "\"field_security\": {" +
                "\"grant\": [\"f1\", \"f2\", \"f3\", \"f4\"]," +
                "\"except\": [\"f3\",\"f4\"]" +
                "}}]}";
        RoleDescriptor rd =
                RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(),
                new String[] { "f1", "f2", "f3", "f4" });
        assertArrayEquals(rd.getIndicesPrivileges()[0].getDeniedFields(),
                new String[] { "f3", "f4" });

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
                "\"field_security\": {" +
                "\"except\": [\"f3\",\"f4\"]," +
                "\"grant\": [\"f1\", \"f2\", \"f3\", \"f4\"]" +
                "}}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(),
                new String[] { "f1", "f2", "f3", "f4" });
        assertArrayEquals(rd.getIndicesPrivileges()[0].getDeniedFields(),
                new String[] { "f3", "f4" });

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
                "\"field_security\": {" +
                "\"grant\": [\"f1\", \"f2\"]" +
                "}}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(),
                new String[] { "f1", "f2" });
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
                "\"field_security\": {" +
                "\"grant\": []" +
                "}}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(), new String[] {});
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
                "\"field_security\": {" +
                "\"except\": []," +
                "\"grant\": []" +
                "}}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(), new String[] {});
        assertArrayEquals(rd.getIndicesPrivileges()[0].getDeniedFields(), new String[] {});

        final String exceptWithoutGrant = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\":" +
                " [\"p3\"], \"field_security\": {" +
                "\"except\": [\"f1\"]" +
                "}}]}";
        ElasticsearchParseException e = expectThrows(ElasticsearchParseException.class,
                () -> RoleDescriptor.parse("test", new BytesArray(exceptWithoutGrant), false,
                        XContentType.JSON));
        assertThat(e.getDetailedMessage(),
                containsString("failed to parse indices privileges for role [test]. field_security"
                        + " requires grant if except is given"));

        final String grantNull = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"]," +
                " \"field_security\": {" +
                "\"grant\": null" +
                "}}]}";
        e = expectThrows(ElasticsearchParseException.class,
                () -> RoleDescriptor.parse("test", new BytesArray(grantNull), false,
                        XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("failed to parse indices privileges for" +
                " role [test]. grant must not be null."));

        final String exceptNull = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": " +
                "[\"p3\"], \"field_security\": {" +
                "\"grant\": [\"*\"]," +
                "\"except\": null" +
                "}}]}";
        e = expectThrows(ElasticsearchParseException.class,
                () -> RoleDescriptor.parse("test", new BytesArray(exceptNull), false,
                        XContentType.JSON));
        assertThat(e.getDetailedMessage(),
                containsString("failed to parse indices privileges for role [test]. except must" +
                        " not be null."));

        final String exceptGrantNull = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": " +
                "[\"p3\"], \"field_security\": {" +
                "\"grant\": null," +
                "\"except\": null" +
                "}}]}";
        e = expectThrows(ElasticsearchParseException.class,
                () -> RoleDescriptor.parse("test", new BytesArray(exceptGrantNull), false,
                        XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("failed to parse indices privileges " +
                "for role [test]. grant must not be null."));

        final String bothFieldsMissing = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": " +
                "[\"p3\"], \"field_security\": {" +
                "}}]}";
        e = expectThrows(ElasticsearchParseException.class,
                () -> RoleDescriptor.parse("test", new BytesArray(bothFieldsMissing), false,
                        XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("failed to parse indices privileges " +
                "for role [test]. \"field_security\" must not be empty."));

        // try with two indices and mix order a little
        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
                "\"field_security\": {" +
                "\"grant\": []" +
                "}}," +
                "{\"names\": \"idx3\",\n" +
                " \"field_security\": {\n" +
                " \"grant\": [\"*\"], \n" +
                " \"except\": [\"f2\"]}," +
            "\"privileges\": [\"p3\"]}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(), new String[] {});
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());
        assertArrayEquals(rd.getIndicesPrivileges()[1].getGrantedFields(), new String[] {"*"});
        assertArrayEquals(rd.getIndicesPrivileges()[1].getDeniedFields(), new String[] {"f2"});
    }

    // test old syntax for field permissions
    public void testBWCFieldPermissions() throws Exception {
        String q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
            "\"fields\": [\"f1\", \"f2\"]" +
            "}]}";
        RoleDescriptor rd = RoleDescriptor.parse("test", new BytesArray(q), true,
            XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(),
            new String[]{"f1", "f2"});
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());

        final String failingQuery = q;
        ElasticsearchParseException e = expectThrows(ElasticsearchParseException.class,
            () -> RoleDescriptor.parse("test", new BytesArray(failingQuery), false,
                XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("[\"fields\": [...]] format has " +
            "changed for field permissions in role [test]" +
            ", use [\"field_security\": {\"grant\":[...],\"except\":[...]}] instead"));

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
            "\"fields\": []" +
            "}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), true, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(), new String[]{});
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());
        final String failingQuery2 = q;
        e = expectThrows(ElasticsearchParseException.class,
            () -> RoleDescriptor.parse("test", new BytesArray(failingQuery2), false,
                XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("[\"fields\": [...]] format has " +
            "changed for field permissions in role [test]" +
            ", use [\"field_security\": {\"grant\":[...],\"except\":[...]}] instead"));

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
            "\"fields\": null" +
            "}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), true, XContentType.JSON);
        assertNull(rd.getIndicesPrivileges()[0].getGrantedFields());
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());
        final String failingQuery3 = q;
        e = expectThrows(ElasticsearchParseException.class,
            () -> RoleDescriptor.parse("test", new BytesArray(failingQuery3), false,
                XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("[\"fields\": [...]] format has " +
            "changed for field permissions in role [test]" +
            ", use [\"field_security\": {\"grant\":[...],\"except\":[...]}] instead"));
    }

    public void testFieldPermissionsHashCodeThreadSafe() throws Exception {
        final int numThreads = scaledRandomIntBetween(4, 16);
        final FieldPermissions fieldPermissions =new FieldPermissions(
            new FieldPermissionsDefinition(new String[] { "*" }, new String[] { "foo" }));
        final CountDownLatch latch = new CountDownLatch(numThreads + 1);
        final AtomicReferenceArray<Integer> hashCodes = new AtomicReferenceArray<>(numThreads);
        List<Thread> threads = new ArrayList<>(numThreads);
        for (int i = 0; i < numThreads; i++) {
            final int threadNum = i;
            threads.add(new Thread(() -> {
            latch.countDown();
            try {
                latch.await();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
            final int hashCode = fieldPermissions.hashCode();
            hashCodes.set(threadNum, hashCode);
            }));
        }

        for (Thread thread : threads) {
            thread.start();
        }
        latch.countDown();
        for (Thread thread : threads) {
            thread.join();
        }

        final int hashCode = fieldPermissions.hashCode();
        for (int i = 0; i < numThreads; i++) {
            assertEquals((Integer) hashCode, hashCodes.get(i));
        }
    }

    public void testAllFieldIsAutomaticallyExcludedIfNotExplicitlyGranted() throws Exception {
        final FieldPermissions fieldPermissions = new FieldPermissions(
            new FieldPermissionsDefinition(new String[] { "_a*" }, new String[0]));
        assertTrue(fieldPermissions.grantsAccessTo("_animal"));
        assertFalse(fieldPermissions.grantsAccessTo("_all"));
```

```
  }

  public void testAllFieldIsNotExcludedIfExplicitlyGranted() throws Exception {
    final String[] grant = { "foo", "bar", "baz", "_all" };
    Collections.shuffle(Arrays.asList(grant), random());

    final FieldPermissions fieldPermissions = new FieldPermissions(
        new FieldPermissionsDefinition(grant, new String[0]));

    assertTrue(fieldPermissions.grantsAccessTo("_all"));
    assertTrue(fieldPermissions.grantsAccessTo("foo"));
    assertTrue(fieldPermissions.grantsAccessTo("bar"));
    assertTrue(fieldPermissions.grantsAccessTo("baz"));
    assertFalse(fieldPermissions.grantsAccessTo(randomAlphaOfLengthBetween(5, 8)));
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.apache.http.message.BasicHeader;
import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.action.DocWriteResponse;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.action.admin.cluster.node.stats.NodesStatsResponse;
import org.elasticsearch.action.admin.cluster.stats.ClusterStatsIndices;
import org.elasticsearch.action.admin.cluster.stats.ClusterStatsResponse;
import org.elasticsearch.action.admin.indices.stats.IndicesStatsResponse;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.client.Client;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.ResponseException;
import org.elasticsearch.client.transport.NoNodeAvailableException;
import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.cluster.routing.ShardRoutingState;
import org.elasticsearch.common.network.NetworkModule;
import org.elasticsearch.common.settings.SecureString;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.util.concurrent.ThreadContext;
import org.elasticsearch.discovery.DiscoveryModule;
import org.elasticsearch.node.MockNode;
import org.elasticsearch.node.Node;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.rest.RestStatus;
import org.elasticsearch.test.SecurityIntegTestCase;
import org.elasticsearch.test.SecuritySettingsSource;
import org.elasticsearch.test.SecuritySettingsSourceField;
import org.elasticsearch.test.discovery.TestZenDiscovery;
import org.elasticsearch.test.junit.annotations.TestLogging;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.transport.Transport;
import org.elasticsearch.xpack.core.TestXPackTransportClient;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.security.SecurityField;
import org.elasticsearch.xpack.core.security.action.user.GetUsersResponse;
import org.elasticsearch.xpack.core.security.authc.support.UsernamePasswordToken;
import org.elasticsearch.xpack.core.security.client.SecurityClient;
import org.elasticsearch.xpack.security.LocalStateSecurity;
import org.junit.After;
import org.junit.Before;

import java.nio.file.Files;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;

import static org.elasticsearch.common.xcontent.XContentFactory.jsonBuilder;
import static org.elasticsearch.test.hamcrest.ElasticsearchAssertions.assertNoFailures;
import static org.hamcrest.Matchers.containsString;
import static org.hamcrest.Matchers.greaterThanOrEqualTo;
import static org.hamcrest.Matchers.hasItem;
import static org.hamcrest.Matchers.is;
import static org.hamcrest.Matchers.notNullValue;

@TestLogging("org.elasticsearch.cluster.service:TRACE,org.elasticsearch.discovery.zen:TRACE,org.elasticsearch.
action.search:TRACE," +
  "org.elasticsearch.search:TRACE")
public class LicensingTests extends SecurityIntegTestCase {
  public static final String ROLES =
        SecuritySettingsSource.TEST_ROLE + ":\n" +
               "  cluster: [ all ]\n" +
               "  indices:\n" +
               "    - names: '*'\n" +
               "      privileges: [manage]\n" +
               "    - names: '/.*/'\n" +
               "      privileges: [write]\n" +
               "    - names: 'test'\n" +
               "      privileges: [read]\n" +
               "    - names: 'test1'\n" +
               "      privileges: [read]\n" +
               "\n" +
               "role_a:\n" +
               "  indices:\n" +
               "    - names: 'a'\n" +
               "      privileges: [all]\n" +
               "\n" +
               "role_b:\n" +
               "  indices:\n" +
               "    - names: 'b'\n" +
               "      privileges: [all]\n";

  public static final String USERS =
        SecuritySettingsSource.CONFIG_STANDARD_USER +
               "user_a:{plain}passwd\n" +
               "user_b:{plain}passwd\n";

  public static final String USERS_ROLES =
        SecuritySettingsSource.CONFIG_STANDARD_USER_ROLES +
               "role_a:user_a,user_b\n" +
               "role_b:user_b\n";

  @Override
  protected String configRoles() {
        return ROLES;
  }

  @Override
  protected String configUsers() {
        return USERS;
  }

  @Override
  protected String configUsersRoles() {
        return USERS_ROLES;
  }

  @Override
  public Settings nodeSettings(int nodeOrdinal) {
        return Settings.builder().put(super.nodeSettings(nodeOrdinal))
               .put(NetworkModule.HTTP_ENABLED.getKey(), true)
        .put(TestZenDiscovery.USE_MOCK_PINGS.getKey(), false)
               .build();
  }

  @Override
  protected Collection<Class<? extends Plugin>> nodePlugins() {
        ArrayList<Class<? extends Plugin>> plugins = new ArrayList<>(super.nodePlugins());
        plugins.add(Netty4Plugin.class); // for http
        return plugins;
  }
  @Override
  protected int maxNumberOfNodes() {
        return super.maxNumberOfNodes() + 1;
  }

  @Before
  public void resetLicensing() {
        enableLicensing();
  }

  @After
  public void cleanupSecurityIndex() {
        deleteSecurityIndex();
  }

  public void testEnableDisableBehaviour() throws Exception {
        IndexResponse indexResponse = index("test", "type", jsonBuilder()
               .startObject()
               .field("name", "value")
               .endObject());
        assertEquals(DocWriteResponse.Result.CREATED, indexResponse.getResult());


        indexResponse = index("test1", "type", jsonBuilder()
               .startObject()
               .field("name", "value1")
               .endObject());
        assertEquals(DocWriteResponse.Result.CREATED, indexResponse.getResult());

        refresh();
        // wait for all replicas to be started (to make sure that there are no more cluster state updates when we disable
licensing)
        assertBusy(() -> assertTrue(client().admin().cluster().prepareState().get().getState().routingTable()
               .shardsWithState(ShardRoutingState.INITIALIZING).isEmpty()));

        Client client = internalCluster().transportClient();

        disableLicensing();

        assertElasticsearchSecurityException(() -> client.admin().indices().prepareStats().get());
        assertElasticsearchSecurityException(() -> client.admin().cluster().prepareClusterStats().get());
        assertElasticsearchSecurityException(() -> client.admin().cluster().prepareHealth().get());
        assertElasticsearchSecurityException(() -> client.admin().cluster().prepareNodesStats().get());

        enableLicensing(randomFrom(License.OperationMode.values()));

        IndicesStatsResponse indicesStatsResponse = client.admin().indices().prepareStats().get();
        assertNoFailures(indicesStatsResponse);

        ClusterStatsResponse clusterStatsNodeResponse = client.admin().cluster().prepareClusterStats().get();
        assertThat(clusterStatsNodeResponse, notNullValue());
        ClusterStatsIndices indices = clusterStatsNodeResponse.getIndicesStats();
        assertThat(indices, notNullValue());
        assertThat(indices.getIndexCount(), greaterThanOrEqualTo(2));

        ClusterHealthResponse clusterIndexHealth = client.admin().cluster().prepareHealth().get();
        assertThat(clusterIndexHealth, notNullValue());

        NodesStatsResponse nodeStats = client.admin().cluster().prepareNodesStats().get();
        assertThat(nodeStats, notNullValue());
    }

    public void testRestAuthenticationByLicenseType() throws Exception {
        Response response = getRestClient().performRequest("GET", "/");
        // the default of the licensing tests is basic
        assertThat(response.getStatusLine().getStatusCode(), is(200));
        ResponseException e = expectThrows(ResponseException.class,
                () -> getRestClient().performRequest("GET", "/_xpack/security/_authenticate"));
        assertThat(e.getResponse().getStatusLine().getStatusCode(), is(403));

        // generate a new license with a mode that enables auth
        License.OperationMode mode = randomFrom(License.OperationMode.GOLD,
License.OperationMode.TRIAL,
                License.OperationMode.PLATINUM, License.OperationMode.STANDARD);
        enableLicensing(mode);
        e = expectThrows(ResponseException.class, () -> getRestClient().performRequest("GET", "/"));
        assertThat(e.getResponse().getStatusLine().getStatusCode(), is(401));
        e = expectThrows(ResponseException.class,
            () -> getRestClient().performRequest("GET", "/_xpack/security/_authenticate"));
        assertThat(e.getResponse().getStatusLine().getStatusCode(), is(401));

        final String basicAuthValue =
UsernamePasswordToken.basicAuthHeaderValue(SecuritySettingsSource.TEST_USER_NAME,
                new SecureString(SecuritySettingsSourceField.TEST_PASSWORD.toCharArray()));
        response = getRestClient().performRequest("GET", "/", new BasicHeader("Authorization", basicAuthValue));
        assertThat(response.getStatusLine().getStatusCode(), is(200));
        response = getRestClient().performRequest("GET", "/_xpack/security/_authenticate",
                    new BasicHeader("Authorization", basicAuthValue));
        assertThat(response.getStatusLine().getStatusCode(), is(200));

    }

    public void testSecurityActionsByLicenseType() throws Exception {
        // security actions should not work!
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateSecurity.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            new SecurityClient(client).prepareGetUsers().get();
            fail("security actions should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
        }

        // enable a license that enables security
        License.OperationMode mode = randomFrom(License.OperationMode.GOLD,
License.OperationMode.TRIAL,
                License.OperationMode.PLATINUM, License.OperationMode.STANDARD);
        enableLicensing(mode);
        // security actions should not work!
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateSecurity.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            GetUsersResponse response = new SecurityClient(client).prepareGetUsers().get();
            assertNotNull(response);
        }
    }

    public void testTransportClientAuthenticationByLicenseType() throws Exception {
        Settings.Builder builder = Settings.builder()
                .put(internalCluster().transportClient().settings());
        // remove user info
        builder.remove(SecurityField.USER_SETTING.getKey());
        builder.remove(ThreadContext.PREFIX + "." + UsernamePasswordToken.BASIC_AUTH_HEADER);

        // basic has no auth
        try (TransportClient client = new TestXPackTransportClient(builder.build(), LocalStateSecurity.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            assertGreenClusterState(client);
        }

        // enable a license that enables security
        License.OperationMode mode = randomFrom(License.OperationMode.GOLD,
License.OperationMode.TRIAL,
                License.OperationMode.PLATINUM, License.OperationMode.STANDARD);
        enableLicensing(mode);

        try (TransportClient client = new TestXPackTransportClient(builder.build(), LocalStateSecurity.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            client.admin().cluster().prepareHealth().get();
```

```
      fail("should not have been able to connect to a node!");
    } catch (NoNodeAvailableException e) {
      // expected
    }
  }

  public void testNodeJoinWithoutSecurityExplicitlyEnabled() throws Exception {
    License.OperationMode mode = randomFrom(License.OperationMode.GOLD,
License.OperationMode.PLATINUM, License.OperationMode.STANDARD);
    enableLicensing(mode);
    ensureGreen();

    Path home = createTempDir();
    Path conf = home.resolve("config");
    Files.createDirectories(conf);
    Settings nodeSettings = Settings.builder()
      .put(nodeSettings(maxNumberOfNodes() - 1).filter(s -> "xpack.security.enabled".equals(s) == false))
      .put("node.name", "my-test-node")
      .put("network.host", "localhost")
      .put("cluster.name", internalCluster().getClusterName())
      .put("discovery.zen.minimum_master_nodes",
        internalCluster().getInstance(Settings.class).get("discovery.zen.minimum_master_nodes"))
      .put("path.home", home)
      .put(TestZenDiscovery.USE_MOCK_PINGS.getKey(), false)
      .put(DiscoveryModule.DISCOVERY_TYPE_SETTING.getKey(), "test-zen")
      .put(DiscoveryModule.DISCOVERY_HOSTS_PROVIDER_SETTING.getKey(), "test-zen")
      .build();
    Collection<Class<? extends Plugin>> mockPlugins = Arrays.asList(LocalStateSecurity.class,
TestZenDiscovery.TestPlugin.class);
    try (Node node = new MockNode(nodeSettings, mockPlugins)) {
      node.start();
      ensureStableCluster(cluster().size() + 1);
    }
  }

  private static void assertElasticsearchSecurityException(ThrowingRunnable runnable) {
    ElasticsearchSecurityException ee = expectThrows(ElasticsearchSecurityException.class, runnable);
    assertThat(ee.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.SECURITY));
    assertThat(ee.status(), is(RestStatus.FORBIDDEN));
  }

  public static void disableLicensing() {
    disableLicensing(License.OperationMode.BASIC);
  }

  public static void disableLicensing(License.OperationMode operationMode) {
    for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
            licenseState.update(operationMode, false);
        }
    }

    public static void enableLicensing() {
        enableLicensing(License.OperationMode.BASIC);
    }

    public static void enableLicensing(License.OperationMode operationMode) {
        for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
            licenseState.update(operationMode, true);
        }
    }
}
```

Copyright (c) 2002-2016, the original author or authors.
All rights reserved.

http://www.opensource.org/licenses/bsd-license.php

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 JLine 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.

# Jackson JSON processor

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

## Licensing

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

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
==
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
==
This product includes software developed by
Joda.org (http://www.joda.org/).

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

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

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

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

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

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

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

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

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

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

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

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

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

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

END OF TERMS AND CONDITIONS

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

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

Copyright 1999-2005 The Apache Software Foundation

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

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

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

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(c) You must retain, in the Source form of any Derivative Works

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

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

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

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

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

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

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache log4j
Copyright 2007 The Apache Software Foundation

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

```
[role="xpack"]
[testenv="basic"]
[[delete-license]]
=== Delete License API

This API enables you to delete licensing information.

[float]
==== Request

`DELETE /_xpack/license`

[float]
==== Description

When your license expires, {xpack} operates in a degraded mode.  For more
information, see {xpack-ref}/license-expiration.html[License Expiration].

[float]
==== Authorization

You must have `manage` cluster privileges to use this API.
For more information, see
{xpack-ref}/security-privileges.html[Security Privileges].

[float]
==== Examples

The following example queries the info API:

[source,js]
---------------------------------------------------------
DELETE _xpack/license
---------------------------------------------------------
// CONSOLE
// TEST[skip:license testing issues]

When the license is successfully deleted, the API returns the following response:
```

```
[source,js]
------------------------------------------------------------
{
 "acknowledged": true
}
------------------------------------------------------------
// NOTCONSOLE
[role="xpack"]
[testenv="basic"]
[[get-license]]
=== Get License API

This API enables you to retrieve licensing information.

[float]
==== Request

`GET /_xpack/license`

[float]
==== Description

This API returns information about the type of license, when it was issued, and
when it expires, for example.

For more information about the different types of licenses, see
https://www.elastic.co/subscriptions.


[float]
==== Query Parameters

`local`::
 (boolean) Specifies whether to retrieve local information. The default value
 is `false`, which means the information is retrieved from the master node.


[float]
==== Authorization

You must have `monitor` cluster privileges to use this API.
For more information, see
{xpack-ref}/security-privileges.html[Security Privileges].


[float]
==== Examples
```

The following example provides information about a basic license:

```
[source,js]
--------------------------------------------------
GET _xpack/license
--------------------------------------------------
// CONSOLE

[source,js]
--------------------------------------------------
{
 "license" : {
   "status" : "active",
   "uid" : "cbff45e7-c553-41f7-ae4f-9205eabd80xx",
   "type" : "basic",
   "issue_date" : "2018-02-22T23:12:05.550Z",
   "issue_date_in_millis" : 1519341125550,
   "max_nodes" : 1000,
   "issued_to" : "test",
   "issuer" : "elasticsearch",
   "start_date_in_millis" : -1
 }
}
--------------------------------------------------
// TESTRESPONSE[s/"cbff45e7-c553-41f7-ae4f-9205eabd80xx"/$body.license.uid/]
// TESTRESPONSE[s/"basic"/$body.license.type/]
// TESTRESPONSE[s/"2018-02-22T23:12:05.550Z"/$body.license.issue_date/]
// TESTRESPONSE[s/1519341125550/$body.license.issue_date_in_millis/]
// TESTRESPONSE[s/1000/$body.license.max_nodes/]
// TESTRESPONSE[s/"test"/$body.license.issued_to/]
// TESTRESPONSE[s/"elasticsearch"/$body.license.issuer/]
[role="xpack"]
[testenv="basic"]
[[update-license]]
=== Update License API

This API enables you to update your license.

[float]
==== Request

`PUT _xpack/license`

[float]
==== Description

You can update your license at runtime without shutting down your nodes.
License updates take effect immediately.  If the license you are installing does
```

not support all of the features that were available with your previous license, however, you are notified in the response. You must then re-submit the API request with the `acknowledge` parameter set to `true`.

For more information about the different types of licenses, see https://www.elastic.co/subscriptions.

[float]
==== Query Parameters

`acknowledge`::
 (boolean) Specifies whether you acknowledge the license changes. The default
 value is `false`.

[float]
==== Request Body

`licenses`::
 (array) A sequence of one or more JSON documents containing the license
 information.

[float]
==== Authorization

If {security} is enabled, you need `manage` cluster privileges to install the license.

If {security} is enabled and you are installing a gold or platinum license, you must enable TLS on the transport networking layer before you install the license. See <<configuring-tls>>.

[float]
==== Examples

The following example updates to a basic license:

[source,js]

```
------------------------------------------------------------
POST _xpack/license
{
 "licenses": [
   {
     "uid":"893361dc-9749-4997-93cb-802e3d7fa4xx",
     "type":"basic",
     "issue_date_in_millis":1411948800000,
     "expiry_date_in_millis":1914278399999,
     "max_nodes":1,
    "issued_to":"issuedTo",
    "issuer":"issuer",
    "signature":"xx"
   }
   ]
}
------------------------------------------------------------
// CONSOLE
// TEST[skip:license testing issues]
```

NOTE: These values are invalid; you must substitute the appropriate content
from your license file.

You can alternatively use a `curl` command, for example:

```
[source,js]
[source,shell]
------------------------------------------------------------
curl -XPUT -u <user> 'http://<host>:<port>/_xpack/license' -H "Content-Type: application/json" -d @license.json
------------------------------------------------------------
// NOTCONSOLE
```

On Windows machine, use the following command:

```
[source,shell]
------------------------------------------------------------
gc .\license.json |  Invoke-WebRequest -uri http://<host>:<port>/_xpack/license -Credential elastic -Method Put -ContentType "application/json"
------------------------------------------------------------
```

In these examples,

* `<user>` is a user ID with the appropriate authority.
* `<host>` is the hostname of the {es} node (`localhost` if executing
 locally)
* `<port>` is the http port (defaults to `9200`)
* `license.json` is the license JSON file

NOTE: If your {es} node has SSL enabled on the HTTP interface, you must
 start your URL with `https://`

If you previously had a license with more features than the basic license, you
receive the following response:

```
[source,js]
------------------------------------------------------------
 {
 "acknowledged": false,
 "license_status": "valid",
 "acknowledge": {
   "message": """This license update requires acknowledgement. To acknowledge the license, please read the following messages and update the license again, this time with the "acknowledge=true" parameter:""",
   "watcher": [
     "Watcher will be disabled"
   ],
   "logstash": [
     "Logstash will no longer poll for centrally-managed pipelines"
   ],
   "security": [
     "The following X-Pack security functionality will be disabled: ..." ]
   }
}
------------------------------------------------------------
// NOTCONSOLE
```

To complete the update, you must re-submit the API request and set the `acknowledge` parameter to `true`. For example:

```
[source,js]
------------------------------------------------------------
POST _xpack/license?acknowledge=true
{
 "licenses": [
   {
     "uid":"893361dc-9749-4997-93cb-802e3d7fa4xx",
     "type":"basic",
     "issue_date_in_millis":1411948800000,
     "expiry_date_in_millis":1914278399999,
     "max_nodes":1,
     "issued_to":"issuedTo",
     "issuer":"issuer",
     "signature":"xx"
   }
   ]
}
------------------------------------------------------------
// CONSOLE
// TEST[skip:license testing issues]
```

Alternatively:

```
[source,sh]
------------------------------------------------------------
curl -XPUT -u elastic 'http://<host>:<port>/_xpack/license?acknowledge=true' -H "Content-Type: application/json" -d @license.json
------------------------------------------------------------
```

// NOTCONSOLE

For more information about the features that are disabled when you downgrade
your license, see {xpack-ref}/license-expiration.html[License Expiration].
[role="xpack"]
[[license-settings]]
=== {xpack} License Settings
++++
<titleabbrev>License Settings</titleabbrev>
++++

You can configure this licensing setting in the `elasticsearch.yml` file.
For more information, see
{xpack-ref}/license-management.html[{xpack} License Management].

`xpack.license.self_generated.type`::
Set to `basic` (default) to enable basic {xpack} features. +
+
--
If set to `trial`, the self-generated license gives access only to all the features
of a x-pack for 30 days. You can later downgrade the cluster to a basic license if
needed.
--
This package is free to use under the Elastic license. It contains open source
and free commercial features and access to paid commercial features.
{stack-ov}/license-management.html[Start a 30-day trial] to try out all of the
paid commercial features. See the
https://www.elastic.co/subscriptions[Subscriptions] page for information about
Elastic license levels.
== Copyright and License

This software is Copyright (c) 2013-2018 by Elasticsearch BV.

This is free software, licensed under The Apache License Version 2.0.
== License

Copyright 2013-2018 Elasticsearch

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

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

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

limitations under the License.
/*
The MIT License

Copyright (c) 2004-2015 Paul R. Holser, Jr.

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.
*/
Eclipse Foundation Software User Agreement

April 9, 2014

Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE
PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR
THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE
THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE
AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT
AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY
NOT USE THE CONTENT.

Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms

and
conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is
also available at http://www.eclipse.org/legal/epl-v10.html. For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the Eclipse Foundation source code repository ("Repository") in software modules ("Modules") and made available as downloadable archives ("Downloads").

*   Content may be structured and packaged into modules to facilitate delivering, extending, and upgrading the Content.
    Typical modules may include plug-ins ("Plug-ins"), plug-in fragments ("Fragments"), and features ("Features").
*   Each Plug-in or Fragment may be packaged as a sub-directory or JAR (Java ARchive) in a directory named "plugins".
*   A Feature is a bundle of one or more Plug-ins and/or Fragments and associated material. Each Feature may be packaged
    as a sub-directory in a directory named "features". Within a Feature, files named "feature.xml" may contain a list
    of the names and version numbers of the Plug-ins and/or Fragments associated with that Feature.
*   Features may also include other Features ("Included Features"). Within a Feature, files named "feature.xml" may
    contain a list of the names and version numbers of Included Features.

The terms and conditions governing Plug-ins and Fragments should be contained in files named "about.html" ("Abouts").
The terms and conditions governing Features and Included Features should be contained in files named "license.html"
("Feature Licenses"). Abouts and Feature Licenses may be located in any directory of a Download or Module including, but
not limited to the following locations:

*   The top-level (root) directory
*   Plug-in and Fragment directories
*   Inside Plug-ins and Fragments packaged as JARs
*   Sub-directories of the directory named "src" of certain Plug-ins
*   Feature directories

Note: if a Feature made available by the Eclipse Foundation is installed using the Provisioning Technology (as defined
below), you must agree to a license ("Feature Update License") during the installation process. If the Feature contains
Included Features, the Feature Update License should either provide you with the terms and conditions governing the
Included Features or inform you where you can locate them. Feature Update Licenses may be found in the "license"
property of files named "feature.properties" found within a Feature. Such Abouts, Feature Licenses, and Feature Update
Licenses contain the terms and conditions (or references to such terms and conditions) that govern your use of the
associated Content in that directory.

THE ABOUTS, FEATURE LICENSES, AND FEATURE UPDATE LICENSES MAY REFER TO THE EPL OR OTHER LICENSE AGREEMENTS, NOTICES OR TERMS AND CONDITIONS. SOME OF THESE OTHER LICENSE AGREEMENTS MAY INCLUDE (BUT ARE NOT LIMITED TO):

* Eclipse Distribution License Version 1.0 (available at http://www.eclipse.org/licenses/edl-v10.html)
* Common Public License Version 1.0 (available at http://www.eclipse.org/legal/cpl-v10.html)
* Apache Software License 1.1 (available at http://www.apache.org/licenses/LICENSE)
* Apache Software License 2.0 (available at http://www.apache.org/licenses/LICENSE-2.0)
* Mozilla Public License Version 1.1 (available at http://www.mozilla.org/MPL/MPL-1.1.html)

IT IS YOUR OBLIGATION TO READ AND ACCEPT ALL SUCH TERMS AND CONDITIONS PRIOR TO USE OF THE CONTENT. If no About, Feature License, or Feature Update License is provided, please contact the Eclipse Foundation to determine what terms and conditions govern that particular Content.

### Use of Provisioning Technology

The Eclipse Foundation makes available provisioning software, examples of which include, but are not limited to, p2 and the Eclipse Update Manager ("Provisioning Technology") for the purpose of allowing users to install software, documentation, information and/or other materials (collectively "Installable Software"). This capability is provided with the intent of allowing such users to install, extend and update Eclipse-based products. Information about packaging Installable Software is available at http://eclipse.org/equinox/p2/repository_packaging.html ("Specification").

You may use Provisioning Technology to allow other parties to install Installable Software. You shall be responsible for enabling the applicable license agreements relating to the Installable Software to be presented to, and accepted by, the users of the Provisioning Technology in accordance with the Specification. By using Provisioning Technology in such a manner and making it available in accordance with the Specification, you further acknowledge your agreement to, and the acquisition of all necessary rights to permit the following:

1. A series of actions may occur ("Provisioning Process") in which a user may execute the Provisioning Technology on a machine ("Target Machine") with the intent of installing, extending or updating the functionality of an Eclipse-based product.
2. During the Provisioning Process, the Provisioning Technology may cause third party Installable Software or a portion thereof to be accessed and copied to the Target Machine.
3. Pursuant to the Specification, you will provide to the user the terms and conditions that govern the use of the Installable Software ("Installable Software Agreement") and such Installable Software Agreement shall be accessed from the Target Machine in accordance with the Specification. Such Installable Software Agreement must inform the

user of the terms and conditions that govern the Installable Software and must solicit acceptance by the end user in the manner prescribed in such Installable Software Agreement. Upon such indication of agreement by the user, the provisioning Technology will complete installation of the Installable Software.

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import,
possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries, or both.
About This Content

May 22, 2015

License

The Eclipse Foundation makes available all content in this plug-in ("Content"). Unless otherwise indicated below, the
Content is provided to you under the terms and conditions of the Apache License, Version 2.0. A copy of the Apache
License, Version 2.0 is available at http://www.apache.org/licenses/LICENSE-2.0.txt

If you did not receive this Content directly from the Eclipse Foundation, the Content is being redistributed by another
party ("Redistributor") and different terms and conditions may apply to your use of any object code in the Content.
Check the Redistributors license that was provided with the Content. If no such license exists, contact the
Redistributor. Unless otherwise indicated below, the terms and conditions of the Apache License, Version 2.0 still apply
to any source code in the Content and such source code may be obtained at
http://www.eclipse.org](http://www.eclipse.org.
// --
// Transitive dependencies of this project determined from the
// maven pom organized by organization.
// --

Apache Extras for Apache log4j.

From: 'an unknown organization'
 - geronimo-jms_1.1_spec org.apache.geronimo.specs:geronimo-jms_1.1_spec:jar:1.0

From: 'Apache Software Foundation' (http://www.apache.org)

- Apache Log4j (http://logging.apache.org/log4j/1.2/) log4j:log4j:bundle:1.2.17
License: The Apache Software License, Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0.txt)

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

The code for the t-digest was originally authored by Ted Dunning

A number of small but very helpful changes have been contributed by Adrien Grand (https://github.com/jpountz)

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

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 Eclipse Foundation, 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.

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

source, and configuration files.

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

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

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

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

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

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

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

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

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

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

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

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

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

You may add Your own copyright statement to Your modifications and

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

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

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

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

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

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

defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

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

Copyright 2010-2013, Carrot Search s.c., Boznicza 11/56, Poznan, Poland

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

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

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

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of

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

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

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

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

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

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

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

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

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

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

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

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

has been advised of the possibility of such damages.

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

END OF TERMS AND CONDITIONS

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

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

Copyright {yyyy} {name of copyright owner}

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

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

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

ACKNOWLEDGEMENT
===============

HPPC borrowed code, ideas or both from:

* Apache Lucene, http://lucene.apache.org/
  (Apache license)
* Fastutil, http://fastutil.di.unimi.it/
  (Apache license)

* Koloboke, https://github.com/OpenHFT/Koloboke
  (Apache license)
The code in this repository code was Written by Gil Tene, Michael Barker,
and Matt Warren, and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

For users of this code who wish to consume it under the "BSD" license rather than under the public domain or CC0 contribution text mentioned above, the code found under this directory is *also* provided under the following license (commonly referred to as the BSD 2-Clause License). This license does not detract from the above stated release of the code into the public domain, and simply represents an additional license granted by the Author.

** Beginning of "BSD 2-Clause License" text. **

Copyright (c) 2012, 2013, 2014 Gil Tene
Copyright (c) 2014 Michael Barker
Copyright (c) 2014 Matt Warren
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.

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.
Elasticsearch
Copyright 2009-2018 Elasticsearch

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

ELASTIC LICENSE AGREEMENT

PLEASE READ CAREFULLY THIS ELASTIC LICENSE AGREEMENT (THIS "AGREEMENT"), WHICH CONSTITUTES A LEGALLY BINDING AGREEMENT AND GOVERNS ALL OF YOUR USE OF ALL OF THE ELASTIC SOFTWARE WITH WHICH THIS AGREEMENT IS INCLUDED ("ELASTIC SOFTWARE") THAT IS PROVIDED IN OBJECT CODE FORMAT, AND, IN ACCORDANCE WITH SECTION 2 BELOW, CERTAIN OF THE ELASTIC SOFTWARE THAT IS PROVIDED IN SOURCE CODE FORMAT. BY INSTALLING OR USING ANY OF THE ELASTIC SOFTWARE GOVERNED BY THIS AGREEMENT, YOU ARE ASSENTING TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH SUCH TERMS AND CONDITIONS, YOU MAY NOT INSTALL OR USE THE ELASTIC SOFTWARE GOVERNED BY THIS AGREEMENT. IF YOU ARE INSTALLING OR USING THE SOFTWARE ON BEHALF OF A LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE THE ACTUAL AUTHORITY TO AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT ON BEHALF OF SUCH ENTITY.

Posted Date: April 20, 2018

This Agreement is entered into by and between Elasticsearch BV ("Elastic") and You, or the legal entity on behalf of whom You are acting (as applicable, "You").

1. OBJECT CODE END USER LICENSES, RESTRICTIONS AND THIRD PARTY OPEN SOURCE SOFTWARE

1.1 Object Code End User License. Subject to the terms and conditions of Section 1.2 of this Agreement, Elastic hereby grants to You, AT NO CHARGE and for so long as you are not in breach of any provision of this Agreement, a License to the Basic Features and Functions of the Elastic Software.

1.2 Reservation of Rights; Restrictions. As between Elastic and You, Elastic and its licensors own all right, title and interest in and to the Elastic Software, and except as expressly set forth in Sections 1.1, and 2.1 of this Agreement, no other license to the Elastic Software is granted to You under this Agreement, by implication, estoppel or otherwise. You agree not to: (i) reverse engineer or decompile, decrypt, disassemble or otherwise reduce any Elastic Software provided to You in Object Code, or any portion thereof, to Source Code, except and only to the extent any such restriction is prohibited by applicable law, (ii) except as expressly permitted in this Agreement, prepare derivative works from, modify, copy or use the Elastic Software Object Code or the Commercial Software Source Code in any manner; (iii) except as expressly permitted in Section 1.1 above, transfer, sell, rent, lease, distribute, sublicense, loan or otherwise transfer, Elastic Software Object Code, in whole or in part, to any third party; (iv) use Elastic Software Object Code for providing time-sharing services, any software-as-a-service, service bureau services or as part of an application services provider or other service offering (collectively, "SaaS Offering") where obtaining access to the Elastic Software or the features and functions of the Elastic Software is a primary reason or substantial motivation for users of the SaaS Offering to access and/or use the SaaS Offering ("Prohibited SaaS Offering"); (v) circumvent the limitations on use of Elastic Software provided to You in Object Code format that are imposed or preserved by any License Key, or (vi) alter or remove any Marks and Notices in the Elastic Software. If You have any question as to whether a specific SaaS Offering constitutes a Prohibited SaaS Offering, or are interested in obtaining Elastic's permission to engage in commercial or non-commercial distribution of the Elastic Software, please contact elastic_license@elastic.co.

1.3 Third Party Open Source Software. The Commercial Software may contain or be provided with third party open source libraries, components, utilities and other open source software (collectively, "Open Source Software"), which Open Source Software may have applicable license terms as identified on a website designated by Elastic. Notwithstanding anything to the contrary herein, use of the Open Source Software shall be subject to the license terms and conditions applicable to such Open Source Software, to the extent required by the applicable licensor (which terms shall not restrict the license rights granted to You hereunder, but may contain additional rights). To the extent any condition of this Agreement conflicts with any license to the Open Source Software, the Open Source Software license will govern with respect to such Open Source Software only. Elastic may also separately provide you with certain open source software that is licensed by Elastic. Your use of such Elastic open source software will not be governed by this Agreement, but by the applicable open source license terms.

2. COMMERCIAL SOFTWARE SOURCE CODE

2.1 Limited License. Subject to the terms and conditions of Section 2.2 of this Agreement, Elastic hereby grants to You, AT NO CHARGE and for so long as you are not in breach of any provision of this Agreement, a limited, non-exclusive, non-transferable, fully paid up royalty free right and license to the Commercial Software in Source Code format, without the right to grant or authorize sublicenses, to prepare Derivative Works of the Commercial Software, provided You (i) do not hack the licensing mechanism, or otherwise circumvent the intended limitations on the use of Elastic Software to enable features other than Basic Features and Functions or those features You are entitled to as part of a Subscription, and (ii) use the resulting object code only for reasonable testing purposes.

2.2 Restrictions. Nothing in Section 2.1 grants You the right to (i) use the Commercial Software Source Code other than in accordance with Section 2.1 above, (ii) use a Derivative Work of the Commercial Software outside of a Non-production Environment, in any production capacity, on a temporary or permanent basis, or (iii) transfer, sell, rent, lease, distribute, sublicense, loan or otherwise make available the Commercial Software Source Code, in whole or in part, to any third party. Notwithstanding the foregoing, You may maintain a copy of the repository in which the Source Code of the Commercial Software resides and that copy may be publicly accessible, provided that you include this Agreement with Your copy of the repository.

3. TERMINATION

3.1 Termination. This Agreement will automatically terminate, whether or not You receive notice of such Termination from Elastic, if You breach any of its provisions.

3.2 Post Termination. Upon any termination of this Agreement, for any reason, You shall promptly cease the use of the Elastic Software in Object Code format and cease use of the Commercial Software in Source Code format. For the avoidance of doubt, termination of this Agreement will not affect Your right to use Elastic Software, in either Object Code or Source Code formats, made available under the Apache License Version 2.0.

3.3 Survival. Sections 1.2, 2.2. 3.3, 4 and 5 shall survive any termination or expiration of this Agreement.

4. DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY

4.1 Disclaimer of Warranties. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, THE ELASTIC SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, AND ELASTIC AND ITS LICENSORS MAKE NO WARRANTIES WHETHER EXPRESSED, IMPLIED OR STATUTORY REGARDING OR RELATING TO THE ELASTIC SOFTWARE. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, ELASTIC AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE ELASTIC SOFTWARE, AND WITH RESPECT TO THE USE OF THE FOREGOING. FURTHER, ELASTIC DOES NOT WARRANT RESULTS OF USE OR THAT THE ELASTIC SOFTWARE WILL BE ERROR FREE OR THAT THE USE OF THE ELASTIC SOFTWARE WILL BE UNINTERRUPTED.

4.2 Limitation of Liability. IN NO EVENT SHALL ELASTIC OR ITS LICENSORS BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DIRECT OR INDIRECT DAMAGES, INCLUDING, WITHOUT LIMITATION, FOR ANY LOSS OF PROFITS, LOSS OF USE, BUSINESS INTERRUPTION, LOSS OF DATA, COST OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONNECTION WITH OR ARISING OUT OF THE USE OR INABILITY TO USE THE ELASTIC SOFTWARE, OR THE PERFORMANCE OF OR FAILURE TO PERFORM THIS AGREEMENT, WHETHER ALLEGED AS A BREACH OF CONTRACT OR TORTIOUS CONDUCT, INCLUDING NEGLIGENCE, EVEN IF ELASTIC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

5. MISCELLANEOUS

This Agreement completely and exclusively states the entire agreement of the parties regarding the subject matter herein, and it supersedes, and its terms govern, all prior proposals, agreements, or other communications between the parties, oral or written, regarding such subject matter. This Agreement may be modified by Elastic from time to time, and any such modifications will be effective upon the "Posted Date" set forth at the top of the modified Agreement. If any provision hereof is held unenforceable, this Agreement will continue without said provision and be interpreted to reflect the original intent of the parties. This Agreement and any non-contractual obligation arising out of or in connection with it, is governed exclusively by Dutch law. This Agreement shall not be governed by the 1980 UN Convention on Contracts for the International Sale of Goods. All disputes arising out of or in connection with this Agreement, including its existence and validity, shall be resolved by the courts with jurisdiction in Amsterdam, The Netherlands, except where mandatory law provides for the courts at another location in The Netherlands to have jurisdiction. The parties hereby irrevocably waive any and all claims and defenses either might otherwise have in any such action or proceeding in any of such courts based upon any alleged lack of personal jurisdiction, improper venue, forum non conveniens or any similar claim or defense. A breach or threatened breach, by You of Section 2 may cause irreparable harm for which damages at law may not provide adequate relief, and therefore Elastic shall be entitled to seek injunctive relief without being required to post a bond. You may not assign this Agreement (including by operation of law in connection with a merger or acquisition), in whole or in part to any third party without the prior written consent of Elastic, which may be withheld or granted by Elastic in its sole and absolute discretion. Any assignment in violation of the preceding sentence is void. Notices to Elastic may also be sent to legal@elastic.co.

6. DEFINITIONS

The following terms have the meanings ascribed:

6.1 "Affiliate" means, with respect to a party, any entity that controls, is controlled by, or which is under common control with, such party, where "control" means ownership of at least fifty percent (50%) of the outstanding voting shares of the entity, or the contractual right to establish policy for, and manage the operations of, the entity.

6.2 "Basic Features and Functions" means those features and functions of the Elastic Software that are eligible for use under a Basic license, as set forth at https://www.elastic.co/subscriptions, as may be modified by Elastic from time to time.

6.3 "Commercial Software" means the Elastic Software Source Code in any file containing a header stating the contents are subject to the Elastic License or which is contained in the repository folder labeled "x-pack", unless a LICENSE file present in the directory subtree declares a different license.

6.4 "Derivative Work of the Commercial Software" means, for purposes of this Agreement, any modification(s) or enhancement(s) to the Commercial Software, which represent, as a whole, an original work of authorship.

6.5 "License" means a limited, non-exclusive, non-transferable, fully paid up, royalty free, right and license, without the right to grant or authorize sublicenses, solely for Your internal business operations to (i) install and use the applicable Features and Functions of the Elastic Software in Object Code, and (ii) permit Contractors and Your Affiliates to use the Elastic software as set forth in (i) above, provided that such use by Contractors must be solely for Your benefit and/or the benefit of Your Affiliates, and You shall be responsible for all acts and omissions of such Contractors and Affiliates in connection with their use of the Elastic software that are contrary to the terms and conditions of this Agreement.

6.6 "License Key" means a sequence of bytes, including but not limited to a JSON blob, that is used to enable certain features and functions of the Elastic Software.

6.7 "Marks and Notices" means all Elastic trademarks, trade names, logos and notices present on the Documentation as originally provided by Elastic.

6.8 "Non-production Environment" means an environment for development, testing or quality assurance, where software is not used for production purposes.

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

6.10 "Source Code" means the preferred form of computer software for making modifications, including but not limited to software source code, documentation source, and configuration files.

6.11 "Subscription" means the right to receive Support Services and a License to the Commercial Software.

Source code in this repository is variously licensed under the Apache License Version 2.0, an Apache compatible license, or the Elastic License. Outside of the "x-pack" folder, source code in a given file is licensed under the Apache License Version 2.0, unless otherwise noted at the beginning of the file or a LICENSE file present in the directory subtree declares a separate license. Within the "x-pack" folder, source code in a given file is licensed under the Elastic License, unless otherwise noted at the beginning of the file or a LICENSE file present in the directory subtree declares a separate license.

The build produces two sets of binaries - one set that falls under the Elastic License and another set that falls under Apache License Version 2.0. The binaries that contain `-oss` in the artifact name are licensed under the Apache License Version 2.0.

/*
* Licensed to Elasticsearch under one or more contributor
* license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright

* ownership. Elasticsearch licenses this file to you under
* the Apache License, Version 2.0 (the "License"); you may
* not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
Apache HttpComponents AsyncClient
Copyright 2010-2016 The Apache Software Foundation

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

Apache HttpCore NIO
Copyright 2005-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache Commons Logging
Copyright 2003-2013 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache HttpComponents Core
Copyright 2005-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
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.
Joni is released under the MIT License.

JCodings is released under the MIT License.

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7. Disclaimer of Warranty. Unless required by applicable law or

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

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

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

END OF TERMS AND CONDITIONS

The art of simplicity is a puzzle of complexity.

## Overview ##

[YAML](http://yaml.org) is a data serialization format designed for human readability and interaction with scripting languages.

SnakeYAML is a YAML processor for the Java Virtual Machine.

## SnakeYAML features ##

* a **complete** [YAML 1.1 processor](http://yaml.org/spec/1.1/current.html). In particular, SnakeYAML can parse all examples from the specification.
* Unicode support including UTF-8/UTF-16 input/output.
* high-level API for serializing and deserializing native Java objects.
* support for all types from the [YAML types repository](http://yaml.org/type/index.html).

* relatively sensible error messages.

## Info ##
* [Changes](https://bitbucket.org/asomov/snakeyaml/wiki/Changes)
* [Documentation](https://bitbucket.org/asomov/snakeyaml/wiki/Documentation)

## Contribute ##
* Mercurial DVCS is used to dance with the [source code](https://bitbucket.org/asomov/snakeyaml/src).
* If you find a bug in SnakeYAML, please [file a bug
report](https://bitbucket.org/asomov/snakeyaml/issues?status=new&status=open).
* You may discuss SnakeYAML at
[the mailing list](http://groups.google.com/group/snakeyaml-core).
Apache Commons Codec
Copyright 2002-2015 The Apache Software Foundation

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

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.
Apache Commons CLI
Copyright 2001-2009 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

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

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

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

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

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

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

1.7. License. means this document.

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

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

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

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

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

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

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

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

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

2. License Grants.

2.1. The Initial Developer Grant.

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

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

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

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

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

2.2. Contributor Grant.

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

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

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

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

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

3. Distribution Obligations.

3.1. Availability of Source Code.

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

3.2. Modifications.

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

3.3. Required Notices.

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

3.4. Application of Additional Terms.

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

3.5. Distribution of Executable Versions.

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

3.6. Larger Works.

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

4. Versions of the License.

4.1. New Versions.

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

4.2. Effect of New Versions.

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

4.3. Modified Versions.

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

5. DISCLAIMER OF WARRANTY.

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

6. TERMINATION.

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

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon

60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

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

7. LIMITATION OF LIABILITY.

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

8. U.S. GOVERNMENT END USERS.

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

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the

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

10. RESPONSIBILITY FOR CLAIMS.

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

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

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

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

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

Preamble

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

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

Copyright (C)

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

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

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

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

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

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

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

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

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

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

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

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

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

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

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

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

===

This project contains annotations in the package org.apache.http.annotation
which are derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
Full text: http://creativecommons.org/licenses/by/2.5/legalcode

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
to create and reproduce Derivative Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:
Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.
Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).
Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.
If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied,

and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

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

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Apache HttpComponents Core
Copyright 2005-2014 The Apache Software Foundation

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

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net
Apache HttpComponents Client
Copyright 1999-2015 The Apache Software Foundation

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

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

source, and configuration files.

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

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

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

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

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

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

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

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

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

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

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

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

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

You may add Your own copyright statement to Your modifications and

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

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

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

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

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

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

defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

Apache License
Version 2.0, January 2004

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:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. 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
4. 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

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such other license terms will then apply in lieu of the terms of the software license above.

JSON processing code subject to the JSON License from JSON.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 shall be used for Good, not Evil.

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.
AWS SDK for Java
Copyright 2010-2014 Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by
Amazon Technologies, Inc (http://www.amazon.com/).

****
THIRD PARTY COMPONENTS
****
This software includes third party software subject to the following copyrights:
- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.
- JSON parsing and utility functions from JSON.org - Copyright 2002 JSON.org.
- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The licenses for these third party components are included in LICENSE.txt
Copyright (c) 2004-2014 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the

"Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

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

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

===

== NOTICE file for use with the Apache License, Version 2.0, ==

===

Apache JAMES Mime4j
Copyright 2004-2010 The Apache Software Foundation

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

This product test suite includes data (mimetools-testmsgs folder) developed by Eryq and ZeeGee Software Inc as part of the "MIME-tools" Perl5 toolkit and licensed under the Artistic License

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

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

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

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

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

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

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

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

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

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

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

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

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

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

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

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

APACHE POI SUBCOMPONENTS:

Apache POI includes subcomponents with separate copyright notices and
license terms. Your use of these subcomponents is subject to the terms
and conditions of the following licenses:

Office Open XML schemas (ooxml-schemas-1.1.jar)

The Office Open XML schema definitions used by Apache POI are
a part of the Office Open XML ECMA Specification (ECMA-376, [1]).
As defined in section 9.4 of the ECMA bylaws [2], this specification
is available to all interested parties without restriction:

> 9.4 All documents when approved shall be made available to
> all interested parties without restriction.

Furthermore, both Microsoft and Adobe have granted patent licenses
to this work [3,4,5].

[1] http://www.ecma-international.org/publications/standards/Ecma-376.htm
[2] http://www.ecma-international.org/memento/Ecmabylaws.htm
[3] http://www.microsoft.com/openspecifications/en/us/programs/osp/default.aspx
[4] http://www.ecma-international.org/publications/files/ECMA-ST/Ecma%20PATENT/Patent%20statements%20ok/ECMA-376%20Edition%202%20Microsoft%20Patent%20Declaration.pdf
[5] http://www.ecma-international.org/publications/files/ECMA-ST/Ecma%20PATENT/Patent%20statements%20ok/ECMA-376%20Adobe%20Patent%20Declaration.pdf

JUnit test library (junit-4.11.jar)

Common Public License - v 1.0

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

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and
documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

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

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

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

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

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

2. GRANT OF RIGHTS

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

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

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

3. REQUIREMENTS

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

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of
the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

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

4. COMMERCIAL DISTRIBUTION

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

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

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED

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. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), 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 OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

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

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

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

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Apache PDFBox
Copyright 2014 The Apache Software Foundation

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

Based on source code originally developed in the PDFBox and
FontBox projects.

Copyright (c) 2002-2007, www.pdfbox.org

Based on source code originally developed in the PaDaF project.
Copyright (c) 2010 Atos Worldline SAS

Includes the Adobe Glyph List
Copyright 1997, 1998, 2002, 2007, 2010 Adobe Systems Incorporated.

Includes the Zapf Dingbats Glyph List
Copyright 2002, 2010 Adobe Systems Incorporated.

Includes OSXAdapter
Copyright (C) 2003-2007 Apple, Inc., All Rights Reserved

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

THIS PRODUCT ALSO INCLUDES THIRD PARTY SOFTWARE REDISTRIBUTED UNDER THE FOLLOWING LICENSES:

Apache Commons Logging,
The Apache Software License, Version 1.1 (commons-logging-1.1.1.jar)

The Apache Software License, Version 1.1

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

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

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

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:
"This product includes software developed by the
Apache Software Foundation (http://www.apache.org/)."
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION 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.

Test messages from the Perl-MIME-Tools project,

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

Licensing of XZ for Java
=======================

All the files in this package have been written by Lasse Collin
and/or Igor Pavlov. All these files have been put into the
public domain. You can do whatever you want with these files.

This software is provided "as is", without any warranty.
Apache Tika
Copyright 2015 The Apache Software Foundation

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

Copyright 1993-2010 University Corporation for Atmospheric Research/Unidata
This software contains code derived from UCAR/Unidata's NetCDF library.

Tika-server component uses CDDL-licensed dependencies: jersey (http://jersey.java.net/) and
Grizzly (http://grizzly.java.net/)

Tika-parsers component uses CDDL/LGPL dual-licensed dependency: jhighlight
(https://github.com/codelibs/jhighlight)

OpenCSV: Copyright 2005 Bytecode Pty Ltd. Licensed under the Apache License, Version 2.0

IPTC Photo Metadata descriptions Copyright 2010 International Press Telecommunications Council.

Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

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

```
   =========================================================================
  ==  NOTICE file corresponding to section 4(d) of the Apache License,   ==
  ==  Version 2.0, in this case for the Apache XmlBeans distribution.    ==
  =========================================================================
```

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

Portions of this software were originally based on the following:
- software copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/>.

Aside from contributions to the Apache XMLBeans project, this
software also includes:

- one or more source files from the Apache Xerces-J and Apache Axis
  products, Copyright (c) 1999-2003 Apache Software Foundation

- W3C XML Schema documents Copyright 2001-2003 (c) World Wide Web
  Consortium (Massachusetts Institute of Technology, European Research
  Consortium for Informatics and Mathematics, Keio University)

- resolver.jar from Apache Xml Commons project,
  Copyright (c) 2001-2003 Apache Software Foundation

- Piccolo XML Parser for Java from http://piccolo.sourceforge.net/,
  Copyright 2002 Yuval Oren under the terms of the Apache Software License 2.0

- JSR-173 Streaming API for XML from http://sourceforge.net/projects/xmlpullparser/,
  Copyright 2005 BEA under the terms of the Apache Software License 2.0

The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(http://www.bouncycastle.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.

MOZILLA PUBLIC LICENSE
Version 1.1

----

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

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

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

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

1.8. "License" means this document.

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

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

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

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

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

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

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

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

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

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of

Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
(a) Third Party Claims.
If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights

granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.
If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.
Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code,

and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

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

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised

and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

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

8. TERMINATION.

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

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY

CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

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

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (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.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is ______________________________________.

The Initial Developer of the Original Code is ________________________.
Portions created by ______________________ are Copyright (C) ______
_______________________. All Rights Reserved.

Contributor(s): ______________________________________.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [______] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

Copyright (c) 2006-2007, www.jempbox.org
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. Neither the name of fontbox; 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 REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Commons Compress
Copyright 2002-2015 The Apache Software Foundation

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

The files in the package org.apache.commons.compress.archivers.sevenz
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

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

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

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

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

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

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

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

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

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

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

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

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

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

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

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

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

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

END OF TERMS AND CONDITIONS

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

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

Copyright [yyyy] [name of copyright owner]

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

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

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

APACHE TIKA SUBCOMPONENTS

Apache Tika includes a number of subcomponents with separate copyright notices
and license terms. Your use of these subcomponents is subject to the terms and
conditions of the following licenses.

MIME type information from file-4.26.tar.gz (http://www.darwinsys.com/file/)

Copyright (c) Ian F. Darwin 1986, 1987, 1989, 1990, 1991, 1992, 1994, 1995.
Software written by Ian F. Darwin and others;
maintained 1994- Christos Zoulas.

This software is not subject to any export provision of the United States
Department of Commerce, and may be exported to any country or planet.

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 immediately at the beginning of the file, without modification,
   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 AUTHOR 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 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.

Charset detection code from ICU4J (http://site.icu-project.org/)

Copyright (c) 1995-2009 International Business Machines Corporation
and others

All rights reserved.

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

copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

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 COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR 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.

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.

Parsing functionality provided by the NetCDF Java Library (http://www.unidata.ucar.edu/software/netcdf-java/)

Copyright 1993-2010 University Corporation for Atmospheric Research/Unidata

Portions of this software were developed by the Unidata Program at the University Corporation for Atmospheric Research.

Access and use of this software shall impose the following obligations and understandings on the user. The user is granted the right, without any fee or cost, to use, copy, modify, alter, enhance and distribute this software, and any derivative works thereof, and its supporting documentation for any purpose whatsoever, provided that this entire notice appears in all copies of the software, derivative works and supporting documentation. Further, UCAR requests that the user credit UCAR/Unidata in any publications that result from the use of this software or in any product that includes this software, although this is not an obligation. The names UCAR and/or Unidata, however, may not be used in any advertising or publicity to endorse or promote any products or commercial entity unless specific written permission is obtained from UCAR/Unidata. The user also understands that UCAR/Unidata is not obligated to provide the user with any support, consulting, training or assistance of any kind with regard to the use, operation and performance of this software nor to provide the user with any updates, revisions, new versions or "bug fixes."

THIS SOFTWARE IS PROVIDED BY UCAR/UNIDATA "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 UCAR/UNIDATA 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 ACCESS,
USE OR PERFORMANCE OF THIS SOFTWARE.

IPTC Photo Metadata descriptions are taken from the IPTC Photo Metadata
Standard, July 2010, Copyright 2010 International Press Telecommunications
Council.

1. The Specifications and Materials are licensed for use only on the condition that you agree to be bound by the terms of this license. Subject to this and other licensing requirements contained herein, you may, on a non-exclusive basis, use the Specifications and Materials.
2. The IPTC openly provides the Specifications and Materials for voluntary use by individuals, partnerships, companies, corporations, organizations and any other entity for use at the entity's own risk. This disclaimer, license and release is intended to apply to the IPTC, its officers, directors, agents, representatives, members, contributors, affiliates, contractors, or co-venturers acting jointly or severally.
3. The Document and translations thereof may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the copyright and license notices and references to the IPTC appearing in the Document and the terms of this Specifications License Agreement are included on all such copies and derivative works. Further, upon the receipt of written permission from the IPTC, the Document may be modified for the purpose of developing applications that use IPTC Specifications or as required to translate the Document into languages other than English.
4. Any use, duplication, distribution, or exploitation of the Document and Specifications and Materials in any manner is at your own risk.
5. NO WARRANTY, EXPRESSED OR IMPLIED, IS MADE REGARDING THE ACCURACY, ADEQUACY, COMPLETENESS, LEGALITY, RELIABILITY OR USEFULNESS OF ANY INFORMATION CONTAINED IN THE DOCUMENT OR IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE PRODUCED OR SPONSORED BY THE IPTC. THE DOCUMENT AND THE INFORMATION CONTAINED HEREIN AND INCLUDED IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE OF THE IPTC IS PROVIDED ON AN "AS IS" BASIS. THE IPTC DISCLAIMS ALL WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY ACTUAL OR ASSERTED WARRANTY OF NON-INFRINGEMENT OF PROPRIETARY RIGHTS, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER THE IPTC NOR ITS CONTRIBUTORS SHALL BE HELD LIABLE FOR ANY IMPROPER OR INCORRECT USE OF INFORMATION. NEITHER THE IPTC NOR ITS CONTRIBUTORS ASSUME ANY RESPONSIBILITY FOR ANYONE'S USE OF INFORMATION PROVIDED BY THE IPTC. IN NO EVENT SHALL THE IPTC OR ITS CONTRIBUTORS BE LIABLE TO ANYONE FOR DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, COMPENSATORY DAMAGES, LOST PROFITS, LOST DATA OR ANY FORM OF SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL OR PUNITIVE DAMAGES OF ANY KIND WHETHER BASED ON BREACH OF CONTRACT OR WARRANTY, TORT, PRODUCT LIABILITY OR OTHERWISE.
6. The IPTC takes no position regarding the validity or scope of any Intellectual Property or other rights that might be claimed to pertain to the implementation or use of the technology described in the Document or the extent to which any license under such rights might or might not be available. The IPTC does not represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication, assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of the Specifications and Materials, can be obtained from the

Managing Director of the IPTC.

7. By using the Specifications and Materials including the Document in any manner or for any purpose, you release the IPTC from all liabilities, claims, causes of action, allegations, losses, injuries, damages, or detriments of any nature arising from or relating to the use of the Specifications, Materials or any portion thereof. You further agree not to file a lawsuit, make a claim, or take any other formal or informal legal action against the IPTC, resulting from your acquisition, use, duplication, distribution, or exploitation of the Specifications, Materials or any portion thereof. Finally, you hereby agree that the IPTC is not liable for any direct, indirect, special or consequential damages arising from or relating to your acquisition, use, duplication, distribution, or exploitation of the Specifications, Materials or any portion thereof.

8. Specifications and Materials may be downloaded or copied provided that ALL copies retain the ownership, copyright and license notices.

9. Materials may not be edited, modified, or presented in a context that creates a misleading or false impression or statement as to the positions, actions, or statements of the IPTC.

10. The name and trademarks of the IPTC may not be used in advertising, publicity, or in relation to products or services and their names without the specific, written prior permission of the IPTC. Any permitted use of the trademarks of the IPTC, whether registered or not, shall be accompanied by an appropriate mark and attribution, as agreed with the IPTC.

11. Specifications may be extended by both members and non-members to provide additional functionality (Extension Specifications) provided that there is a clear recognition of the IPTC IP and its ownership in the Extension Specifications and the related documentation and provided that the extensions are clearly identified and provided that a perpetual license is granted by the creator of the Extension Specifications for other members and non-members to use the Extension Specifications and to continue extensions of the Extension Specifications. The IPTC does not waive any of its rights in the Specifications and Materials in this context. The Extension Specifications may be considered the intellectual property of their creator. The IPTC expressly disclaims any responsibility for damage caused by an extension to the Specifications.

12. Specifications and Materials may be included in derivative work of both members and non-members provided that there is a clear recognition of the IPTC IP and its ownership in the derivative work and its related documentation. The IPTC does not waive any of its rights in the Specifications and Materials in this context. Derivative work in its entirety may be considered the intellectual property of the creator of the work .The IPTC expressly disclaims any responsibility for damage caused when its IP is used in a derivative context.

13. This Specifications License Agreement is perpetual subject to your conformance to the terms of this Agreement. The IPTC may terminate this Specifications License Agreement immediately upon your breach of this Agreement and, upon such termination you will cease all use, duplication, distribution, and/or exploitation in any manner of the Specifications and Materials.

14. This Specifications License Agreement reflects the entire agreement of the parties regarding the subject matter hereof and supersedes all prior agreements or representations regarding such matters, whether written or oral. To the extent any portion or provision of this Specifications License Agreement is found to be illegal or unenforceable, then the remaining provisions of this Specifications License Agreement will remain in full force and effect and the illegal or unenforceable provision will be construed to give it such effect as it may properly have that is consistent with the intentions of the parties.

15. This Specifications License Agreement may only be modified in writing signed by an authorized representative of the IPTC.

16. This Specifications License Agreement is governed by the law of United Kingdom, as such law is applied to contracts made and fully performed in the United Kingdom. Any disputes arising from or relating to this Specifications License Agreement will be resolved in the courts of the United Kingdom. You consent to the jurisdiction of such courts over you and covenant not to assert before such courts any objection to proceeding in such forums.

JUnRAR (https://github.com/edmund-wagner/junrar/)

JUnRAR is based on the UnRAR tool, and covered by the same license
It was formerly available from http://java-unrar.svn.sourceforge.net/

```
******    *****   ******   UnRAR - free utility for RAR archives
**   **  **   **  **   **  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
******   *******  ******    License for use and distribution of
**   **  **   **  **   **   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
**   **  **   **  **   **         FREE portable version
                                  ~~~~~~~~~~~~~~~~~~~~~
```

The source code of UnRAR utility is freeware. This means:

1. All copyrights to RAR and the utility UnRAR are exclusively
owned by the author - Alexander Roshal.

2. The UnRAR sources may be used in any software to handle RAR
archives without limitations free of charge, but cannot be used
to re-create the RAR compression algorithm, which is proprietary.
Distribution of modified UnRAR sources in separate form or as a
part of other software is permitted, provided that it is clearly
stated in the documentation and source comments that the code may
not be used to develop a RAR (WinRAR) compatible archiver.

3. The UnRAR utility may be freely distributed. It is allowed
to distribute UnRAR inside of other software packages.

4. THE RAR ARCHIVER AND THE UnRAR UTILITY ARE DISTRIBUTED "AS IS".
NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT
YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS,
DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING
OR MISUSING THIS SOFTWARE.

5. Installing and using the UnRAR utility signifies acceptance of
these terms and conditions of the license.

6. If you don't agree with terms of the license you must remove
UnRAR files from your storage devices and cease to use the
utility.

Thank you for your interest in RAR and UnRAR. Alexander L. Roshal

Sqlite (bundled in org.xerial's sqlite-jdbc)
This product bundles Sqlite, which is in the Public Domain. For details
see: https://www.sqlite.org/copyright.html

Apache POI
Copyright 2003-2015 The Apache Software Foundation

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

This product contains parts that were originally based on software from BEA.
Copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/>.

This product contains W3C XML Schema documents. Copyright 2001-2003 (c)
World Wide Web Consortium (Massachusetts Institute of Technology, European
Research Consortium for Informatics and Mathematics, Keio University)

This product contains the Piccolo XML Parser for Java
(http://piccolo.sourceforge.net/). Copyright 2002 Yuval Oren.

This product contains the chunks_parse_cmds.tbl file from the vsdump program.
Copyright (C) 2006-2007 Valek Filippov (frob@df.ru)

This product contains parts of the eID Applet project
(http://eid-applet.googlecode.com). Copyright (c) 2009-2014
FedICT (federal ICT department of Belgium), e-Contract.be BVBA (https://www.e-contract.be),
Bart Hanssens from FedICT
The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(http://www.bouncycastle.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.

Apache License
Version 2.0, January 2004

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

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

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

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

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

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

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

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

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

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

EXTERNAL COMPONENTS

Apache PDFBox includes a number of components with separate copyright notices and license terms. Your use of these components is subject to the terms and conditions of the following licenses.

Contributions made to the original PDFBox and FontBox projects:

Copyright (c) 2002-2007, www.pdfbox.org
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. Neither the name of pdfbox; 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 REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Adobe Font Metrics (AFM) for PDF Core 14 Fonts

This file and the 14 PostScript(R) AFM files it accompanies may be used,

copied, and distributed for any purpose and without charge, with or without
modification, provided that all copyright notices are retained; that the
AFM files are not distributed without this file; that all modifications
to this file or any of the AFM files are prominently noted in the modified
file(s); and that this paragraph is not modified. Adobe Systems has no
responsibility or obligation to support the use of the AFM files.

CMaps for PDF Fonts (http://opensource.adobe.com/wiki/display/cmap/Downloads)

Copyright 1990-2009 Adobe Systems Incorporated.
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 Adobe Systems Incorporated 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.

PaDaF PDF/A preflight (http://sourceforge.net/projects/padaf)

Copyright 2010 Atos Worldline SAS

Licensed by Atos Worldline SAS under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
Atos Worldline SAS licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

OSXAdapter

Version: 2.0

Disclaimer: IMPORTANT: This Apple software is supplied to you by
Apple Inc. ("Apple") in consideration of your agreement to the
following terms, and your use, installation, modification or
redistribution of this Apple software constitutes acceptance of these
terms. If you do not agree with these terms, please do not use,
install, modify or redistribute this Apple software.

In consideration of your agreement to abide by the following terms, and
subject to these terms, Apple grants you a personal, non-exclusive
license, under Apple's copyrights in this original Apple software (the
"Apple Software"), to use, reproduce, modify and redistribute the Apple
Software, with or without modifications, in source and/or binary forms;
provided that if you redistribute the Apple Software in its entirety and
without modifications, you must retain this notice and the following
text and disclaimers in all such redistributions of the Apple Software.
Neither the name, trademarks, service marks or logos of Apple Inc.
may be used to endorse or promote products derived from the Apple
Software without specific prior written permission from Apple. Except
as expressly stated in this notice, no other rights or licenses, express
or implied, are granted by Apple herein, including but not limited to
any patent rights that may be infringed by your derivative works or by
other works in which the Apple Software may be incorporated.

The Apple Software is provided by Apple on an "AS IS" basis. APPLE
MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED

AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2003-2007 Apple, Inc., All Rights Reserved

Apache Commons Logging

Copyright 2003-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache Commons IO

Copyright 2002-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This software is Copyright (c) 2013 by MaxMind, Inc.

This is free software, licensed under the Apache License, Version 2.0.

This software is Copyright (c) 2014 by MaxMind, Inc.

This is free software, licensed under the Apache License, Version 2.0.

Copyright (c) 2007-2009, JSR305 expert group
All rights reserved.

http://www.opensource.org/licenses/bsd-license.php

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 JSR305 expert group 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.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Commons Lang
Copyright 2001-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())
Copyright (c) 2004-2014 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Commons Codec
Copyright 2002-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

==

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.
Apache Commons Lang
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())
Copyright (c) <YEAR>, <OWNER>
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.

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.
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0 1.

Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following: A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications; B. Any new file that contains any part of the Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source

Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY. COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS

LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS. The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained

within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.
Apache HTrace (incubating) is Apache 2.0 Licensed. See below for licensing
of dependencies that are NOT Apache Licensed.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Units, a string formatting go library, is Copyright (c) 2014 Alec Thomas
and MIT licensed: https://github.com/alecthomas/units/blob/master/COPYING

D3, a javascript library for manipulating data, used by htrace-hbase
is Copyright 2010-2014, Michael Bostock and BSD licensed:
https://github.com/mbostock/d3/blob/master/LICENSE

Bootstrap, an html, css, and javascript framework, is
Copyright (c) 2011-2015 Twitter, Inc and MIT licensed:
https://github.com/twbs/bootstrap/blob/master/LICENSE

underscore, a javascript library of functional programming helpers, is
(c) 2009-2014 Jeremy Ashkenas, DocumentCloud and Investigative Reporters
& Editors and an MIT license:
https://github.com/jashkenas/underscore/blob/master/LICENSE

jquery, a javascript library, is Copyright jQuery Foundation and other
contributors, https://jquery.org/. The software consists of
voluntary contributions made by many individuals. For exact
contribution history, see the revision history
available at https://github.com/jquery/jquery
It is MIT licensed:
https://github.com/jquery/jquery/blob/master/LICENSE.txt

backbone, is a javascript library, that is Copyright (c) 2010-2014
Jeremy Ashkenas, DocumentCloud. It is MIT licensed:
https://github.com/jashkenas/backbone/blob/master/LICENSE

moment.js is a front end time conversion project.
It is (c) 2011-2014 Tim Wood, Iskren Chernev, Moment.js contributors
and shared under the MIT license:
https://github.com/moment/moment/blob/develop/LICENSE

CMP is an implementation of the MessagePack serialization format in
C. It is licensed under the MIT license:
https://github.com/camgunz/cmp/blob/master/LICENSE

go-codec is an implementation of several serialization and deserialization
codecs in Go. It is licensed under the MIT license:
https://github.com/ugorji/go/blob/master/LICENSE
This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).
Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache HTrace
Copyright 2015 The Apache Software Foundation

This product includes software developed at The Apache Software
Foundation (http://www.apache.org/).

In addition, this product includes software dependencies. See
the accompanying LICENSE.txt for a listing of dependencies
that are NOT Apache licensed (with pointers to their licensing)

Apache HTrace includes an Apache Thrift connector to Zipkin. Zipkin
is a distributed tracing system that is Apache 2.0 Licensed.
Copyright 2012 Twitter, Inc.

Our Owl logo we took from http://www.clker.com/clipart-13653.html.
It is public domain/free.

Apache Commons Configuration
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2006 Envoi Solutions LLC

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.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. "Executable" means the Covered Software in any form other than Source Code.

1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the

Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted

interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL

Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.
ICU4J, (under lucene/analysis/icu) is licensed under an MIT style license
(modules/analysis/icu/lib/icu4j-LICENSE-BSD_LIKE.txt) and Copyright (c) 1995-2012
International Business Machines Corporation and others
ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2012 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

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 COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR 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.

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.

All trademarks and registered trademarks mentioned herein are the property of
their respective owners.
Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Copyright: Elasticsearch B.V. <info@elastic.co>
License: ${license.name}
${license.text}

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

ELASTIC LICENSE AGREEMENT

PLEASE READ CAREFULLY THIS ELASTIC LICENSE AGREEMENT (THIS "AGREEMENT"), WHICH CONSTITUTES A LEGALLY BINDING AGREEMENT AND GOVERNS ALL OF YOUR USE OF ALL OF THE ELASTIC SOFTWARE WITH WHICH THIS AGREEMENT IS INCLUDED ("ELASTIC SOFTWARE") THAT IS PROVIDED IN OBJECT CODE FORMAT, AND, IN ACCORDANCE WITH SECTION 2 BELOW, CERTAIN OF THE ELASTIC SOFTWARE THAT IS PROVIDED IN SOURCE CODE FORMAT. BY INSTALLING OR USING ANY OF THE ELASTIC SOFTWARE GOVERNED BY THIS AGREEMENT, YOU ARE ASSENTING TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH SUCH TERMS AND CONDITIONS, YOU MAY NOT INSTALL OR USE THE ELASTIC SOFTWARE GOVERNED BY THIS AGREEMENT. IF YOU ARE INSTALLING OR USING THE SOFTWARE ON BEHALF OF A LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE THE ACTUAL AUTHORITY TO AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT ON BEHALF OF SUCH ENTITY.

Posted Date: April 20, 2018

This Agreement is entered into by and between Elasticsearch BV ("Elastic") and You, or the legal entity on behalf of whom You are acting (as applicable, "You").

1. OBJECT CODE END USER LICENSES, RESTRICTIONS AND THIRD PARTY OPEN SOURCE SOFTWARE

1.1 Object Code End User License. Subject to the terms and conditions of Section 1.2 of this Agreement, Elastic hereby grants to You, AT NO CHARGE and for so long as you are not in breach of any provision of this Agreement, a License to the Basic Features and Functions of the Elastic Software.

1.2 Reservation of Rights; Restrictions. As between Elastic and You, Elastic and its licensors own all right, title and interest in and to the Elastic Software, and except as expressly set forth in Sections 1.1, and 2.1 of this

Agreement, no other license to the Elastic Software is granted to You under this Agreement, by implication, estoppel or otherwise. You agree not to: (i) reverse engineer or decompile, decrypt, disassemble or otherwise reduce any Elastic Software provided to You in Object Code, or any portion thereof, to Source Code, except and only to the extent any such restriction is prohibited by applicable law, (ii) except as expressly permitted in this Agreement, prepare derivative works from, modify, copy or use the Elastic Software Object Code or the Commercial Software Source Code in any manner; (iii) except as expressly permitted in Section 1.1 above, transfer, sell, rent, lease, distribute, sublicense, loan or otherwise transfer, Elastic Software Object Code, in whole or in part, to any third party; (iv) use Elastic Software Object Code for providing time-sharing services, any software-as-a-service, service bureau services or as part of an application services provider or other service offering (collectively, "SaaS Offering") where obtaining access to the Elastic Software or the features and functions of the Elastic Software is a primary reason or substantial motivation for users of the SaaS Offering to access and/or use the SaaS Offering ("Prohibited SaaS Offering"); (v) circumvent the limitations on use of Elastic Software provided to You in Object Code format that are imposed or preserved by any License Key, or (vi) alter or remove any Marks and Notices in the Elastic Software. If You have any question as to whether a specific SaaS Offering constitutes a Prohibited SaaS Offering, or are interested in obtaining Elastic's permission to engage in commercial or non-commercial distribution of the Elastic Software, please contact elastic_license@elastic.co.

1.3 Third Party Open Source Software. The Commercial Software may contain or be provided with third party open source libraries, components, utilities and other open source software (collectively, "Open Source Software"), which Open Source Software may have applicable license terms as identified on a website designated by Elastic. Notwithstanding anything to the contrary herein, use of the Open Source Software shall be subject to the license terms and conditions applicable to such Open Source Software, to the extent required by the applicable licensor (which terms shall not restrict the license rights granted to You hereunder, but may contain additional rights). To the extent any condition of this Agreement conflicts with any license to the Open Source Software, the Open Source Software license will govern with respect to such Open Source Software only. Elastic may also separately provide you with certain open source software that is licensed by Elastic. Your use of such Elastic open source software will not be governed by this Agreement, but by the applicable open source license terms.

2. COMMERCIAL SOFTWARE SOURCE CODE

2.1 Limited License. Subject to the terms and conditions of Section 2.2 of this Agreement, Elastic hereby grants to You, AT NO CHARGE and for so long as you are not in breach of any provision of this Agreement, a limited, non-exclusive, non-transferable, fully paid up royalty free right and license to the Commercial Software in Source Code format, without the right to grant or authorize sublicenses, to prepare Derivative Works of the Commercial Software, provided You (i) do not hack the licensing mechanism, or otherwise circumvent the intended limitations on the use of Elastic Software to enable features other than Basic Features and Functions or those features You are entitled to as part of a Subscription, and (ii) use the resulting object code only for reasonable testing purposes.

2.2 Restrictions. Nothing in Section 2.1 grants You the right to (i) use the Commercial Software Source Code other than in accordance with Section 2.1 above, (ii) use a Derivative Work of the Commercial Software outside of a Non-production Environment, in any production capacity, on a temporary or permanent basis, or (iii) transfer, sell, rent, lease, distribute, sublicense, loan or otherwise make available the Commercial Software Source Code, in whole or in part, to any third party. Notwithstanding the foregoing, You may maintain a copy of the repository in which the Source Code of the Commercial Software resides and that copy may be publicly accessible, provided that you include this Agreement with Your copy of the repository.

3. TERMINATION

3.1 Termination. This Agreement will automatically terminate, whether or not You receive notice of such Termination from Elastic, if You breach any of its provisions.

3.2 Post Termination. Upon any termination of this Agreement, for any reason, You shall promptly cease the use of the Elastic Software in Object Code format and cease use of the Commercial Software in Source Code format. For the avoidance of doubt, termination of this Agreement will not affect Your right to use Elastic Software, in either Object Code or Source Code formats, made available under the Apache License Version 2.0.

3.3 Survival. Sections 1.2, 2.2. 3.3, 4 and 5 shall survive any termination or expiration of this Agreement.

4. DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY

4.1 Disclaimer of Warranties. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, THE ELASTIC SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, AND ELASTIC AND ITS LICENSORS MAKE NO WARRANTIES WHETHER EXPRESSED, IMPLIED OR STATUTORY REGARDING OR RELATING TO THE ELASTIC SOFTWARE. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, ELASTIC AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE ELASTIC SOFTWARE, AND WITH RESPECT TO THE USE OF THE FOREGOING. FURTHER, ELASTIC DOES NOT WARRANT RESULTS OF USE OR THAT THE ELASTIC SOFTWARE WILL BE ERROR FREE OR THAT THE USE OF THE ELASTIC SOFTWARE WILL BE UNINTERRUPTED.

4.2 Limitation of Liability. IN NO EVENT SHALL ELASTIC OR ITS LICENSORS BE

LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DIRECT OR INDIRECT DAMAGES, INCLUDING, WITHOUT LIMITATION, FOR ANY LOSS OF PROFITS, LOSS OF USE, BUSINESS INTERRUPTION, LOSS OF DATA, COST OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONNECTION WITH OR ARISING OUT OF THE USE OR INABILITY TO USE THE ELASTIC SOFTWARE, OR THE PERFORMANCE OF OR FAILURE TO PERFORM THIS AGREEMENT, WHETHER ALLEGED AS A BREACH OF CONTRACT OR TORTIOUS CONDUCT, INCLUDING NEGLIGENCE, EVEN IF ELASTIC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

5. MISCELLANEOUS

This Agreement completely and exclusively states the entire agreement of the parties regarding the subject matter herein, and it supersedes, and its terms govern, all prior proposals, agreements, or other communications between the parties, oral or written, regarding such subject matter. This Agreement may be modified by Elastic from time to time, and any such modifications will be effective upon the "Posted Date" set forth at the top of the modified Agreement. If any provision hereof is held unenforceable, this Agreement will continue without said provision and be interpreted to reflect the original intent of the parties. This Agreement and any non-contractual obligation arising out of or in connection with it, is governed exclusively by Dutch law. This Agreement shall not be governed by the 1980 UN Convention on Contracts for the International Sale of Goods. All disputes arising out of or in connection with this Agreement, including its existence and validity, shall be resolved by the courts with jurisdiction in Amsterdam, The Netherlands, except where mandatory law provides for the courts at another location in The Netherlands to have jurisdiction. The parties hereby irrevocably waive any and all claims and defenses either might otherwise have in any such action or proceeding in any of such courts based upon any alleged lack of personal jurisdiction, improper venue, forum non conveniens or any similar claim or defense. A breach or threatened breach, by You of Section 2 may cause irreparable harm for which damages at law may not provide adequate relief, and therefore Elastic shall be entitled to seek injunctive relief without being required to post a bond. You may not assign this Agreement (including by operation of law in connection with a merger or acquisition), in whole or in part to any third party without the prior written consent of Elastic, which may be withheld or granted by Elastic in its sole and absolute discretion. Any assignment in violation of the preceding sentence is void. Notices to Elastic may also be sent to legal@elastic.co.

6. DEFINITIONS

The following terms have the meanings ascribed:

6.1 "Affiliate" means, with respect to a party, any entity that controls, is controlled by, or which is under common control with, such party, where "control" means ownership of at least fifty percent (50%) of the outstanding voting shares of the entity, or the contractual right to establish policy for, and manage the operations of, the entity.

6.2 "Basic Features and Functions" means those features and functions of the Elastic Software that are eligible for use under a Basic license, as set forth at https://www.elastic.co/subscriptions, as may be modified by Elastic from time to time.

6.3 "Commercial Software" means the Elastic Software Source Code in any file containing a header stating the contents are subject to the Elastic License or which is contained in the repository folder labeled "x-pack", unless a LICENSE file present in the directory subtree declares a different license.

6.4 "Derivative Work of the Commercial Software" means, for purposes of this Agreement, any modification(s) or enhancement(s) to the Commercial Software, which represent, as a whole, an original work of authorship.

6.5 "License" means a limited, non-exclusive, non-transferable, fully paid up, royalty free, right and license, without the right to grant or authorize sublicenses, solely for Your internal business operations to (i) install and use the applicable Features and Functions of the Elastic Software in Object Code, and (ii) permit Contractors and Your Affiliates to use the Elastic software as set forth in (i) above, provided that such use by Contractors must be solely for Your benefit and/or the benefit of Your Affiliates, and You shall be responsible for all acts and omissions of such Contractors and Affiliates in connection with their use of the Elastic software that are contrary to the terms and conditions of this Agreement.

6.6 "License Key" means a sequence of bytes, including but not limited to a JSON blob, that is used to enable certain features and functions of the Elastic Software.

6.7 "Marks and Notices" means all Elastic trademarks, trade names, logos and notices present on the Documentation as originally provided by Elastic.

6.8 "Non-production Environment" means an environment for development, testing or quality assurance, where software is not used for production purposes.

6.9 "Object Code" means any form resulting from mechanical transformation or translation of Source Code form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

6.10 "Source Code" means the preferred form of computer software for making modifications, including but not limited to software source code, documentation source, and configuration files.

6.11 "Subscription" means the right to receive Support Services and a License to the Commercial Software.

Source code in this repository is variously licensed under the Apache License Version 2.0, an Apache compatible license, or the Elastic License. Outside of the "x-pack" folder, source code in a given file is licensed under the Apache License Version 2.0, unless otherwise noted at the beginning of the file or a LICENSE file present in the directory subtree declares a separate license. Within the "x-pack" folder, source code in a given file is licensed under the Elastic License, unless otherwise noted at the beginning of the file or a LICENSE file present in the directory subtree declares a separate license.

The build produces two sets of binaries - one set that falls under the Elastic License and another set that falls under Apache License Version 2.0. The binaries that contain `-oss` in the artifact name are licensed under the Apache License Version 2.0.
Elasticsearch
Copyright 2009-2018 Elasticsearch

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

1.16 Elasticsearch Rest Client 5.6.2

1.16.1 Available under license :

Elasticsearch
Copyright 2009-2017 Elasticsearch

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

ANTLR 2 License

Antlr2 is released in the public domain.
See licenses/antlr2-license.txt for details.

ASM 4 License

ASM 4 uses a 3-clause BSD license. For details, see licenses/asm-license.txt.
ASM 4 License

Copyright (c) 2000-2011 INRIA, France Telecom
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. Neither the name of the copyright holders 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.
ANTLR 2 License

We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commerical software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

In countries where the Public Domain status of the work may not be valid, the author grants a copyright licence to the general public to deal in the work without restriction and permission to sublicence derivates under the terms of any (OSI approved) Open Source licence.

The Python parser generator code under antlr/actions/python/ is covered by the 3-clause BSD licence (this part is included in the binary JAR files); the run-time part under lib/python/ is covered by the GNU GPL, version 3 or later (this part is not included in the binary JAR files). See [1] for the full details.

https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=750643#80%22
Apache Groovy
Copyright 2003-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes/uses ANTLR (http://www.antlr2.org/)
developed by Terence Parr 1989-2006
Apache Commons Codec
Copyright 2002-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache HttpClient
Copyright 1999-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache HttpAsyncClient
Copyright 2010-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache HttpCore NIO
Copyright 2005-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache Commons Logging
Copyright 2003-2013 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

===

This project contains annotations in the package org.apache.http.annotation
which are derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
Full text: http://creativecommons.org/licenses/by/2.5/legalcode

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.

"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

to create and reproduce Derivative Works;

to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Apache HttpCore
Copyright 2005-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net

The Netty Project
=================

Please visit the Netty web site for more information:

 * http://netty.io/

Copyright 2011 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

 * LICENSE:
   * license/LICENSE.jsr166y.txt (Public Domain)
 * HOMEPAGE:
   * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
   * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

 * LICENSE:
   * license/LICENSE.base64.txt (Public Domain)
 * HOMEPAGE:

  * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified version of 'JZlib', a re-implementation of zlib in pure Java, which can be obtained at:

 * LICENSE:
   * license/LICENSE.jzlib.txt (BSD Style License)
 * HOMEPAGE:
   * http://www.jcraft.com/jzlib/

This product contains a modified version of 'Webbit', a Java event based WebSocket and HTTP server:

 * LICENSE:
   * license/LICENSE.webbit.txt (BSD License)
 * HOMEPAGE:
   * https://github.com/joewalnes/webbit

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

 * LICENSE:
   * license/LICENSE.protobuf.txt (New BSD License)
 * HOMEPAGE:
   * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

 * LICENSE:
   * license/LICENSE.bouncycastle.txt (MIT License)
 * HOMEPAGE:
   * http://www.bouncycastle.org/

This product optionally depends on 'SLF4J', a simple logging facade for Java, which can be obtained at:

 * LICENSE:
   * license/LICENSE.slf4j.txt (MIT License)
 * HOMEPAGE:
   * http://www.slf4j.org/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

 * LICENSE:
   * license/LICENSE.commons-logging.txt (Apache License 2.0)

- HOMEPAGE:
  - http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

- LICENSE:
  - license/LICENSE.log4j.txt (Apache License 2.0)
- HOMEPAGE:
  - http://logging.apache.org/log4j/

This product optionally depends on 'JBoss Logging', a logging framework, which can be obtained at:

- LICENSE:
  - license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
- HOMEPAGE:
  - http://anonsvn.jboss.org/repos/common/common-logging-spi/

This product optionally depends on 'Apache Felix', an open source OSGi framework implementation, which can be obtained at:

- LICENSE:
  - license/LICENSE.felix.txt (Apache License 2.0)
- HOMEPAGE:
  - http://felix.apache.org/

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the

"Work") to the public domain. A certifier, moreover, dedicates any copyright
interest he may have in the associated work, and for these purposes, is
described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this
work. Certifier recognizes that his good faith efforts may not shield him from
liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to
the detriment of the Dedicator's heirs and successors. Dedicator intends this
dedication to be an overt act of relinquishment in perpetuate of all present
and future rights under copyright law, whether vested or contingent, in the
Work. Dedicator understands that such relinquishment of all rights includes
the relinquishment of all rights to enforce (by lawsuit or otherwise) those
copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be
freely reproduced, distributed, transmitted, used, modified, built upon, or
otherwise exploited by anyone for any purpose, commercial or non-commercial,
and in any way, including by methods that have not yet been invented or
conceived.
The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
(http://www.bouncycastle.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.
Copyright (c) 2000,2001,2002,2003,2004 ymnk, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products
   derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/*
* Copyright (c) 2004-2007 QOS.ch
* All rights reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining
* a copy of this software and associated documentation files (the
* "Software"), to deal in the Software without restriction, including
* without limitation the rights to use, copy, modify, merge, publish,
* distribute, sublicense, and/or sell copies of the Software, and to
* permit persons to whom the Software is furnished to do so, subject to
* the following conditions:
*
* The above copyright notice and this permission notice shall be
* included in all copies or substantial portions of the Software.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
* NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
* LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
* OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/
(BSD License: http://www.opensource.org/licenses/bsd-license)

Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
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 Webbit 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.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or

distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN

WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, 5th Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.
Apache Lucene
Copyright 2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, and Hindi analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt

See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers (common) are based on BSD-licensed reference implementations created by Jacques Savoy and Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration

==

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==
mecab-ipadic-2.7.0-20070801 Notice
==

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST 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 tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,

performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the

Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from unicode conversion examples available at http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/
```

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was derived from Brics automaton sources available at www.brics.dk/automaton/. Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
```

```
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``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 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 levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.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.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was

derived from ICU (http://www.icu-project.org)
The full license is available here:
http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* 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 COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR 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.
*
* 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.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
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 copyright holders 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.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER 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 HOLDERS 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 following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
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 Morfologik 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.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,

Marcin Woliski, Robert Woosz

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.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS 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 COPYRIGHT HOLDERS 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.
Elasticsearch
Copyright 2009-2017 Elasticsearch

This product includes software developed by The Apache Software Foundation
(http://www.apache.org/).

==
HdrHistogram LICENSE
==

The code in this repository code was Written by Gil Tene, Michael Barker,
and Matt Warren, and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

For users of this code who wish to consume it under the "BSD" license
rather than under the public domain or CC0 contribution text mentioned
above, the code found under this directory is *also* provided under the
following license (commonly referred to as the BSD 2-Clause License). This
license does not detract from the above stated release of the code into
the public domain, and simply represents an additional license granted by
the Author.

** Beginning of "BSD 2-Clause License" text. **

Copyright (c) 2012, 2013, 2014 Gil Tene
Copyright (c) 2014 Michael Barker
Copyright (c) 2014 Matt Warren
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.

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.

==

antlr4-runtime LICENSE

==

[The "BSD license"]

Copyright (c) 2015 Terence Parr, Sam Harwell
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products

derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``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 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.

==

asm LICENSE

==

Copyright (c) 2012 France Tlcom
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. Neither the name of the copyright holders 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.

==

compiler LICENSE

==

Copyright 2010 RightTime, 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.

==
groovy NOTICE
==

Apache Groovy
Copyright 2003-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

==
groovy LICENSE
==

/*
* 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.
*
*/

==
hppc NOTICE
==

ACKNOWLEDGEMENT
===============

HPPC borrowed code, ideas or both from:

* Apache Lucene, http://lucene.apache.org/
  (Apache license)
* Fastutil, http://fastutil.di.unimi.it/
  (Apache license)
* Koloboke, https://github.com/OpenHFT/Koloboke
  (Apache license)

==
hppc LICENSE
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2010-2013, Carrot Search s.c., Boznicza 11/56, Poznan, Poland

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.

==
jcodings NOTICE
==
JCodings is released under the MIT License.

==
jcodings LICENSE
==
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.

==
jna LICENSE
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

==

joda-time NOTICE

==

```
=============================================================================
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
=============================================================================
```

This product includes software developed by
Joda.org (http://www.joda.org/).

```
================================================================================
joda-time LICENSE
================================================================================
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"

replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
joni NOTICE
==

Joni is released under the MIT License.

==
joni LICENSE
==

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.

==

jopt-simple LICENSE

==

/*
The MIT License

Copyright (c) 2004-2015 Paul R. Holser, Jr.

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.
*/

==

jts LICENSE

==

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser

General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer

system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

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 that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

==
log4j NOTICE
==
Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

==
log4j LICENSE
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==

log4j-api NOTICE

==

Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

==

log4j-api LICENSE

==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
log4j-core NOTICE
==

Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

==
log4j-core LICENSE
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
lucene NOTICE
==

Apache Lucene
Copyright 2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
http://snowball.tartarus.org/

The KStem stemmer in
analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, and Hindi analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,

analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

==

Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration

==

This software includes a binary and/or source version of data from

mecab-ipadic-2.7.0-20070801

which can be obtained from

http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==

mecab-ipadic-2.7.0-20070801 Notice

==

Nara Institute of Science and Technology (NAIST), the copyright holders, disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness, in no event shall NAIST 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 tortuous action, arising out of or in connection with the use or performance of this software.

A large portion of the dictionary entries originate from ICOT Free Software. The following conditions for ICOT Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its original form or modified, to any third party or parties, PROVIDED that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear on, or be attached to, the Program, which is distributed substantially in the same form as set out herein and that such intended distribution, if actually made, will neither violate or otherwise contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express,

implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification, copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

==
lucene LICENSE
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

```
/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer
*
* This source code is provided as is by Unicode, Inc. No claims are
* made as to fitness for any particular purpose. No warranties of any
* kind are expressed or implied. The recipient agrees to determine
* applicability of information provided. If this file has been
* purchased on magnetic or optical media from Unicode, Inc., the
* sole remedy for any claim will be exchange of defective media
* within 90 days of receipt.
*
* Limitations on Rights to Redistribute This Code
*
* Unicode, Inc. hereby grants the right to freely use the information
* supplied in this file in the creation of products supporting the
* Unicode Standard, and to make copies of this file in any form
* for internal or external distribution as long as this notice
* remains attached.
*/
```

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/. Here is the copyright from those sources:

```
/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``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 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 levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.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.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* 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 COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
* LIABLE FOR ANY CLAIM, OR 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.
*
* 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.
*/

The following license applies to the Snowball stemmers:

Copyright (c) 2001, Dr Martin Porter
Copyright (c) 2002, Richard Boulton
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 copyright holders 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.

The following license applies to the KStemmer:

Copyright 2003,
Center for Intelligent Information Retrieval,
University of Massachusetts, Amherst.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and "University of Massachusetts" must not be used to endorse or promote products derived from this software without prior written permission. To obtain permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER 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 HOLDERS 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 following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
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 Morfologik 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.

The dictionary comes from Morfologik project. Morfologik uses data from
Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and
is licenced on the terms of (inter alia) LGPL and Creative Commons
ShareAlike. The part-of-speech tags were added in Morfologik project and
are not found in the data from sjp.pl. The tagset is similar to IPI PAN
tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
Marcin Woliski, Robert Woosz

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.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS 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 COPYRIGHT HOLDERS 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.

==
netty NOTICE
==

The Netty Project
=================

Please visit the Netty web site for more information:

 * http://netty.io/

Copyright 2011 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

 * LICENSE:
   * license/LICENSE.jsr166y.txt (Public Domain)
 * HOMEPAGE:
   * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
   * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

 * LICENSE:
   * license/LICENSE.base64.txt (Public Domain)
 * HOMEPAGE:
   * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified version of 'JZlib', a re-implementation of
zlib in pure Java, which can be obtained at:

 * LICENSE:
   * license/LICENSE.jzlib.txt (BSD Style License)
 * HOMEPAGE:
   * http://www.jcraft.com/jzlib/

This product contains a modified version of 'Webbit', a Java event based
WebSocket and HTTP server:

 * LICENSE:
   * license/LICENSE.webbit.txt (BSD License)

* HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product optionally depends on 'Protocol Buffers', Google's data interchange format, which can be obtained at:

* LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
* HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate a temporary self-signed X.509 certificate when the JVM does not provide the equivalent functionality. It can be obtained at:

* LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
* HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'SLF4J', a simple logging facade for Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
* HOMEPAGE:
  * http://www.slf4j.org/

This product optionally depends on 'Apache Commons Logging', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
* HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework, which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
* HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'JBoss Logging', a logging framework, which can be obtained at:

* LICENSE:

* license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
* HOMEPAGE:
  * http://anonsvn.jboss.org/repos/common/common-logging-spi/

This product optionally depends on 'Apache Felix', an open source OSGi framework implementation, which can be obtained at:

* LICENSE:
  * license/LICENSE.felix.txt (Apache License 2.0)
* HOMEPAGE:
  * http://felix.apache.org/

===
netty LICENSE
===

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==

securesm LICENSE

==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==

snakeyaml NOTICE

==

The art of simplicity is a puzzle of complexity.

## Overview ##
[YAML](http://yaml.org) is a data serialization format designed for human readability and interaction with scripting languages.

SnakeYAML is a YAML processor for the Java Virtual Machine.

## SnakeYAML features ##

* a **complete** [YAML 1.1 processor](http://yaml.org/spec/1.1/current.html). In particular, SnakeYAML can parse all examples from the specification.
* Unicode support including UTF-8/UTF-16 input/output.
* high-level API for serializing and deserializing native Java objects.
* support for all types from the [YAML types repository](http://yaml.org/type/index.html).
* relatively sensible error messages.

## Info ##
* [Changes](https://bitbucket.org/asomov/snakeyaml/wiki/Changes)
* [Documentation](https://bitbucket.org/asomov/snakeyaml/wiki/Documentation)

## Contribute ##
* Mercurial DVCS is used to dance with the [source code](https://bitbucket.org/asomov/snakeyaml/src).
* If you find a bug in SnakeYAML, please [file a bug report](https://bitbucket.org/asomov/snakeyaml/issues?status=new&status=open).
* You may discuss SnakeYAML at [the mailing list](http://groups.google.com/group/snakeyaml-core).

===

snakeyaml LICENSE

===

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

==

spatial4j NOTICE

==

Eclipse Foundation Software User Agreement

April 9, 2014

Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE
PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR
THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE
THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE
AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT
AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY
NOT USE THE CONTENT.

Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and
conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is
also available at http://www.eclipse.org/legal/epl-v10.html. For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the

Eclipse Foundation source code repository ("Repository") in software modules ("Modules") and made available as downloadable archives ("Downloads").

*   Content may be structured and packaged into modules to facilitate delivering, extending, and upgrading the Content.
   Typical modules may include plug-ins ("Plug-ins"), plug-in fragments ("Fragments"), and features ("Features").
*   Each Plug-in or Fragment may be packaged as a sub-directory or JAR (Java ARchive) in a directory named "plugins".
*   A Feature is a bundle of one or more Plug-ins and/or Fragments and associated material. Each Feature may be packaged
   as a sub-directory in a directory named "features". Within a Feature, files named "feature.xml" may contain a list
   of the names and version numbers of the Plug-ins and/or Fragments associated with that Feature.
*   Features may also include other Features ("Included Features"). Within a Feature, files named "feature.xml" may
   contain a list of the names and version numbers of Included Features.

The terms and conditions governing Plug-ins and Fragments should be contained in files named "about.html" ("Abouts").
The terms and conditions governing Features and Included Features should be contained in files named "license.html"
("Feature Licenses"). Abouts and Feature Licenses may be located in any directory of a Download or Module including, but
not limited to the following locations:

*   The top-level (root) directory
*   Plug-in and Fragment directories
*   Inside Plug-ins and Fragments packaged as JARs
*   Sub-directories of the directory named "src" of certain Plug-ins
*   Feature directories

Note: if a Feature made available by the Eclipse Foundation is installed using the Provisioning Technology (as defined
below), you must agree to a license ("Feature Update License") during the installation process. If the Feature contains
Included Features, the Feature Update License should either provide you with the terms and conditions governing the
Included Features or inform you where you can locate them. Feature Update Licenses may be found in the "license"
property of files named "feature.properties" found within a Feature. Such Abouts, Feature Licenses, and Feature Update
Licenses contain the terms and conditions (or references to such terms and conditions) that govern your use of the
associated Content in that directory.

THE ABOUTS, FEATURE LICENSES, AND FEATURE UPDATE LICENSES MAY REFER TO THE EPL OR OTHER LICENSE AGREEMENTS, NOTICES OR
TERMS AND CONDITIONS. SOME OF THESE OTHER LICENSE AGREEMENTS MAY INCLUDE (BUT ARE NOT LIMITED TO):

*   Eclipse Distribution License Version 1.0 (available at http://www.eclipse.org/licenses/edl-v10.html)
*   Common Public License Version 1.0 (available at http://www.eclipse.org/legal/cpl-v10.html)

* Apache Software License 1.1 (available at http://www.apache.org/licenses/LICENSE)
* Apache Software License 2.0 (available at http://www.apache.org/licenses/LICENSE-2.0)
* Mozilla Public License Version 1.1 (available at http://www.mozilla.org/MPL/MPL-1.1.html)

IT IS YOUR OBLIGATION TO READ AND ACCEPT ALL SUCH TERMS AND CONDITIONS PRIOR TO USE OF THE CONTENT. If no About, Feature License, or Feature Update License is provided, please contact the Eclipse Foundation to determine what terms and conditions govern that particular Content.

### Use of Provisioning Technology

The Eclipse Foundation makes available provisioning software, examples of which include, but are not limited to, p2 and the Eclipse Update Manager ("Provisioning Technology") for the purpose of allowing users to install software, documentation, information and/or other materials (collectively "Installable Software"). This capability is provided with the intent of allowing such users to install, extend and update Eclipse-based products. Information about packaging Installable Software is available at http://eclipse.org/equinox/p2/repository_packaging.html ("Specification").

You may use Provisioning Technology to allow other parties to install Installable Software. You shall be responsible for enabling the applicable license agreements relating to the Installable Software to be presented to, and accepted by, the users of the Provisioning Technology in accordance with the Specification. By using Provisioning Technology in such a manner and making it available in accordance with the Specification, you further acknowledge your agreement to, and the acquisition of all necessary rights to permit the following:

1. A series of actions may occur ("Provisioning Process") in which a user may execute the Provisioning Technology on a machine ("Target Machine") with the intent of installing, extending or updating the functionality of an Eclipse-based product.
2. During the Provisioning Process, the Provisioning Technology may cause third party Installable Software or a portion thereof to be accessed and copied to the Target Machine.
3. Pursuant to the Specification, you will provide to the user the terms and conditions that govern the use of the Installable Software ("Installable Software Agreement") and such Installable Software Agreement shall be accessed from the Target Machine in accordance with the Specification. Such Installable Software Agreement must inform the user of the terms and conditions that govern the Installable Software and must solicit acceptance by the end user in the manner prescribed in such Installable Software Agreement. Upon such indication of agreement by the user, the provisioning Technology will complete installation of the Installable Software.

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the

import,
possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries, or both.

==
spatial4j LICENSE
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
t-digest NOTICE
==
The code for the t-digest was originally authored by Ted Dunning

A number of small but very helpful changes have been contributed by Adrien Grand (https://github.com/jpountz)

==
t-digest LICENSE
==
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This copy of Jackson JSON processor YAML module is licensed under the Apache (Software) License, version 2.0 ("the License"). See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

Apache Log4j 1.x Compatibility API
Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
===
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
===
This product includes software developed by
Joda.org (http://www.joda.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache Log4j Core
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell

Apache Log4j API
Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.17 elasticsearch-curator 5.6.0

1.17.1 Available under license :

Copyright 20112017 Elasticsearch <http://elastic.co> and contributors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

The following is a list of people who have contributed ideas, code, bug
reports, or in general have helped curator along its way.

Contributors:
* Jordan Sissel (jordansissel) (For Logstash, first and foremost)
* Shay Banon (kimchy) (For Elasticsearch, of course!)
* Aaron Mildenstein (untergeek)
* Njal Karevoll
* Franois Deppierraz
* Honza Kral (HonzaKral)
* Benjamin Smith (benjaminws)
* Colin Moller (LeftyBC)
* Elliot (edgeofnite)
* Ram Viswanadha (ramv)
* Chris Meisinger (cmeisinger)
* Stuart Warren (stuart-warren)
* (gitshaw)
* (sfritz)
* (sjoelsam)
* Jose Diaz-Gonzalez (josegonzalez)
* Arie Bro (arieb)
* David Harrigan (dharrigan)
* Mathieu Geli (gelim)
* Nick Ethier (nickethier)
* Mohab Usama (mohabusama)
* (gitshaw)
* Stuart Warren (stuart-warren)
* Xavier Calland (xavier-calland)
* Chad Schellenger (cschellenger)
* Kamil Essekkat (ekamil)
* (gbutt)
* Ben Buchacher (bbuchacher)
* Ehtesh Choudhury (shurane)

* Markus Fischer (mfn)
* Fabien Wernli (faxm0dem)
* Michael Weiser (michaelweiser)
* (digital-wonderland)
* cassiano (cassianoleal)
* Matt Dainty (bodgit)
* Alex Philipp (alex-sf)
* (krzaczek)
* Justin Lintz (jlintz)
* Jeremy Falling (jjfalling)
* Ian Babrou (bobrik)
* Ferenc Erki (ferki)
* George Heppner (gheppner)
* Matt Hughes (matthughes)
* Brian Lalor (blalor)
* Pawe Krzaczkowski (krzaczek)
* Ben Tse (bt5e)
* Tom Hendrikx (whyscream)
* Christian Vozar (christianvozar)
* Magnus Baeck (magnusbaeck)
* Robin Kearney (rk295)
* (cfeio)
* (malagoli)
* Dan Sheridan (djs52)
* Michael-Keith Bernard (SegFaultAX)
* Simon Lundstrm (simmel)
* (pkr1234)
* Mark Feltner (feltnerm)
* William Jimenez (wjimenez5271)
* Jeremy Canady (jrmycanady)
* Steven Ottenhoff (steffo)
* Ole Rner (Basster)
* Jack (univerio)
* Tom Mzes (hydrapolic)
* Gary Gao (garyelephant)
* Panagiotis Moustafellos (pmoust)
* (pbamba)
* Pavel Strashkin (xaka)
* Wadim Kruse (wkruse)
* Richard Megginson (richm)
* Thibaut Ackermann (thib-ack)
* (zzugg)
* Julien Mancuso (petitout)
* Spencer Herzberg (sherzberg)
* Luke Waite (lukewaite)
* (dtrv)
* Christopher "Chief" Najewicz (chiefy)
* Filipe Gonalves (basex)

* Snke Liebau (soenkeliebau)
* Timothy Schroder (tschroeder-zendesk)
* Jared Carey (jpcarey)
* Juraj Seffer (jurajseffer)
* Roger Steneteg (rsteneteg)

Copyright 2011"2017 Elasticsearch <http://elastic.co> and contributors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text

file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

1.18 elasticsearch-dsl 6.3.1

1.18.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

1.19 elasticsearch-python 6.3.1

1.19.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
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.

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.

1.20 faker 4.1.0

1.20.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

faker.js - Copyright (c) 2010-2015
Matthew Bergman & Marak Squires
http://github.com/marak/faker.js/

faker.js was inspired by and has used data definitions from:

* https://github.com/stympy/faker/ - Copyright (c) 2007-2010 Benjamin Curtis
* http://search.cpan.org/~jasonk/Data-Faker-0.07/ - Copyright 2004-2005 by Jason Kohles

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.

1.21 faker 4.1.0

1.21.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner

or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions

of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
faker.js - Copyright (c) 2010-2015
Matthew Bergman & Marak Squires
http://github.com/marak/faker.js/

faker.js was inspired by and has used data definitions from:

* https://github.com/stympy/faker/ - Copyright (c) 2007-2010 Benjamin Curtis
* http://search.cpan.org/~jasonk/Data-Faker-0.07/ - Copyright 2004-2005 by Jason Kohles

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.
faker.js - Copyright (c) 2010-2015
Matthew Bergman & Marak Squires
http://github.com/marak/faker.js/

faker.js was inspired by and has used data definitions from:

* https://github.com/stympy/faker/ - Copyright (c) 2007-2010 Benjamin Curtis
* http://search.cpan.org/~jasonk/Data-Faker-0.07/ - Copyright 2004-2005 by Jason Kohles

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.
faker.js - Copyright (c) 2010-2015
Matthew Bergman & Marak Squires
http://github.com/marak/faker.js/

faker.js was inspired by and has used data definitions from:

* https://github.com/stympy/faker/ - Copyright (c) 2007-2010 Benjamin Curtis
* http://search.cpan.org/~jasonk/Data-Faker-0.07/ - Copyright 2004-2005 by Jason Kohles

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.
faker.js - Copyright (c) 2010-2015
Matthew Bergman & Marak Squires
http://github.com/marak/faker.js/

faker.js was inspired by and has used data definitions from:

* https://github.com/stympy/faker/ - Copyright (c) 2007-2010 Benjamin Curtis
* http://search.cpan.org/~jasonk/Data-Faker-0.07/ - Copyright 2004-2005 by Jason Kohles

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.

1.22 Flask 1.0.2

1.22.1 Available under license :

Copyright (c) 2010 by Armin Ronacher.

Some rights reserved.

This logo or a modified version may be used by anyone to refer to the
Flask project, but does not indicate endorsement by the project.

Redistribution and use in source (the SVG file) and binary forms (rendered
PNG files etc.) of the image, with or without modification, are permitted
provided that the following conditions are met:

* Redistributions of source code must retain the above copyright
 notice and this list of conditions.

* The names of the contributors to the Flask software (see AUTHORS) may
 not be used to endorse or promote products derived from this software
 without specific prior written permission.

Note: we would appreciate that you make the image a link to
http://flask.pocoo.org/ if you use it on a web page.
Copyright 2010 by the Pallets team.

Some rights reserved.

Redistribution and use in source and binary forms of the software as

well as documentation, 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 copyright holder 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 AND DOCUMENTATION 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 AND DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
License
=======

Flask is licensed under a three clause BSD License. It basically means:
do whatever you want with it as long as the copyright in Flask sticks
around, the conditions are not modified and the disclaimer is present.
Furthermore you must not use the names of the authors to promote derivatives
of the software without written consent.

The full license text can be found below (:ref:`flask-license`). For the
documentation and artwork different licenses apply.

.. _authors:

Authors

.. include:: ../AUTHORS

General License Definitions

The following section contains the full license texts for Flask and the documentation.

- "AUTHORS" hereby refers to all the authors listed in the :ref:`authors` section.

- The ":ref:`flask-license`" applies to all the source code shipped as part of Flask (Flask itself as well as the examples and the unittests) as well as documentation.

- The ":ref:`artwork-license`" applies to the project's Horn-Logo.

.. _flask-license:

Flask License

.. include:: ../LICENSE

.. _artwork-license:

Flask Artwork License

.. include:: ../artwork/LICENSE

1.23 Flask-Cors 3.0.7

1.23.1 Available under license :

Copyright (C) 2016 Cory Dolphin, Olin College

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) 2016 Cory Dolphin, Olin College

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) 2016 Cory Dolphin, Olin College

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 flink 1.4.2_Scala_2.11

1.24.1 Available under license :

==
NOTICE corresponding to the section 4d of
The Apache License, Version 2.0, for
flink-table
==

The 'flink-table' module bundles certain dependencies into
the binary release artifacts.
Below are the license statements where required by the license.

Janino version (org.codehaus.janino : janino) v. 3.0.7

under the following license:

Janino - An embedded Java[TM] compiler

Copyright (c) 2001-2016, Arno Unkrig
Copyright (c) 2015-2016 TIBCO Software Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of JANINO 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 HOLDERS 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.

Reflections (org.reflections : reflections) v. 0.9.10
under the following license:

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. You just DO WHAT THE FUCK YOU WANT TO.

Apache Flink
Copyright 2014-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

==

Apache Flink contains subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of their respective licenses.

See the LICENSE file for a list of subcomponents and dependencies and their respective licenses.

lodash:

Copyright 2012-2015 The Dojo Foundation <http://dojofoundation.org/>
Based on Underscore.js, copyright 2009-2015 Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

```
/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements.  See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership.  The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License.  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package org.apache.flink.api.common.functions.util;

import java.util.List;
import org.apache.flink.annotation.Internal;
import org.apache.flink.api.common.typeutils.TypeSerializer;
import org.apache.flink.util.Collector;

@Internal
public class CopyingListCollector<T> implements Collector<T> {

 private final List<T> list;
 private final TypeSerializer<T> serializer;

 public CopyingListCollector(List<T> list, TypeSerializer<T> serializer) {
  this.list = list;
  this.serializer = serializer;
 }

 @Override
 public void collect(T record) {
  list.add(serializer.copy(record));
 }

 @Override
 public void close() {}
}
/*
* Licensed to the Apache Software Foundation (ASF) under one
* or more contributor license agreements.  See the NOTICE file
* distributed with this work for additional information
* regarding copyright ownership.  The ASF licenses this file
* to you under the Apache License, Version 2.0 (the
* "License"); you may not use this file except in compliance
* with the License.  You may obtain a copy of the License at
*
*     http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package org.apache.flink.api.common.functions.util;

import java.util.Iterator;

import org.apache.flink.annotation.Internal;
import org.apache.flink.api.common.typeutils.TypeSerializer;
import org.apache.flink.util.TraversableOnceException;
```

```
@Internal
public class CopyingIterator<E> implements Iterator<E>, Iterable<E> {

 private final Iterator<E> source;
 private final TypeSerializer<E> serializer;

 private boolean available = true;

 public CopyingIterator(Iterator<E> source, TypeSerializer<E> serializer) {
  this.source = source;
  this.serializer = serializer;
 }

 @Override
 public Iterator<E> iterator() {
  if (available) {
   available = false;
   return this;
  } else {
   throw new TraversableOnceException();
  }
 }

 @Override
 public boolean hasNext() {
  return source.hasNext();
 }

 @Override
 public E next() {
  E next = source.next();
  return serializer.copy(next);
 }

 @Override
 public void remove() {
  throw new UnsupportedOperationException();
 }
}
```

This project includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project bundles the following dependencies under
the Apache Software License 2.0

- org.apache.lucene : lucene-core version 4.10.4
- org.apache.lucene : lucene-analyzers-common version 4.10.4
- org.apache.lucene : lucene-grouping version 4.10.4
- org.apache.lucene : lucene-highlighter version 4.10.4
- org.apache.lucene : lucene-join version 4.10.4
- org.apache.lucene : lucene-memory version 4.10.4
- org.apache.lucene : lucene-misc version 4.10.4
- org.apache.lucene : lucene-queries version 4.10.4
- org.apache.lucene : lucene-queryparser version 4.10.4
- org.apache.lucene : lucene-sandbox version 4.10.4
- org.apache.lucene : lucene-spatial version 4.10.4
- org.apache.lucene : lucene-suggest version 4.10.4
- com.spatial4j : spatial4j version 0.4.1
- com.fasterxml.jackson.core : jackson-core version 2.5.3
- com.fasterxml.jackson.dataformat : jackson-dataformat-smile version 2.5.3
- com.fasterxml.jackson.dataformat : jackson-dataformat-yaml version 2.5.3
- com.fasterxml.jackson.dataformat : jackson-dataformat-cbor version 2.5.3
- org.joda : joda-convert (copied classes)

===================================

Notice for Yaml

===================================

This project bundles yaml (v. 1.12) under the Creative Commons License (CC-BY 2.0).

Original project website: http://www.yaml.de

Copyright (c) 2005-2013, Dirk Jesse

YAML under Creative Commons License (CC-BY 2.0)
===

The YAML framework is published under the Creative Commons Attribution 2.0 License (CC-BY 2.0), which permits
both private and commercial use (http://creativecommons.org/licenses/by/2.0/).

Condition: For the free use of the YAML framework, a backlink to the YAML homepage (http://www.yaml.de) in a suitable place (e.g.: footer of the website or in the imprint) is required.

===================================

Notice for Tartarus

===================================

This project bundles tartarus under the MIT License.

Original source repository: https://github.com/sergiooramas/tartarus

Copyright (c) 2017 Sergio Oramas and Oriol Nieto

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.

====================================

Notice for Antlr

====================================

This project bundles antlr-runtime (v. 3.5) under the BSD 2-Clause License

Copyright (c) 2010 Terence Parr

** Beginning of "BSD 2-Clause License" text. **

Copyright (c) 2010 Terence Parr
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.

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 project includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project bundles the following dependencies under
the Apache Software License 2.0

- org.apache.lucene : lucene-core version 6.3.0
- org.apache.lucene : lucene-analyzers-common version 6.3.0
- org.apache.lucene : lucene-backward-codecs version 6.3.0
- org.apache.lucene : lucene-grouping version 6.3.0
- org.apache.lucene : lucene-highlighter version 6.3.0
- org.apache.lucene : lucene-join version 6.3.0
- org.apache.lucene : lucene-memory version 6.3.0
- org.apache.lucene : lucene-misc version 6.3.0
- org.apache.lucene : lucene-queries version 6.3.0
- org.apache.lucene : lucene-queryparser version 6.3.0
- org.apache.lucene : lucene-sandbox version 6.3.0
- org.apache.lucene : lucene-spatial version 6.3.0
- org.apache.lucene : lucene-spatial-extras version 6.3.0
- org.apache.lucene : lucene-spatial3d version 6.3.0
- org.apache.lucene : lucene-suggest version 6.3.0
- org.apache.httpcomponents : httpclient version 4.5.3
- org.apache.httpcomponents : httpcore version 4.4.6
- org.apache.httpcomponents : httpasynclcient version 4.1.2
- org.apache.httpcomponents : httpcore-nio version 4.4.5
- com.carrotsearch : hppc version 0.7.1
- com.fasterxml.jackson.core : jackson-core version 2.8.1
- com.fasterxml.jackson.dataformat : jackson-dataformat-smile version 2.8.1
- com.fasterxml.jackson.dataformat : jackson-dataformat-yaml version 2.8.1
- com.fasterxml.jackson.dataformat : jackson-dataformat-cbor version 2.8.1
- com.tdunning : t-digest version 3.0
- io.netty : netty version 3.10.6.Final
- io.netty : netty-buffer version 4.1.6.Final
- io.netty : netty-codec version 4.1.6.Final
- io.netty : netty-codec-http version 4.1.6.Final
- io.netty : netty-common version 4.1.6.Final
- io.netty : netty-handler version 4.1.6.Final
- io.netty : netty-resolver version 4.1.6.Final
- io.netty : netty-transport version 4.1.6.Final
- org.jboss.netty : netty version 3.2.0.Final

- joda-time : joda-time version 2.5
- com.github.spullara.mustache.java : compiler version 0.9.3

===================================

Notice for Yaml

===================================

This project bundles yaml (v. 1.15) under the Creative Commons License (CC-BY 2.0).

Original project website: http://www.yaml.de

Copyright (c) 2005-2013, Dirk Jesse

YAML under Creative Commons License (CC-BY 2.0)
===

The YAML framework is published under the Creative Commons Attribution 2.0 License (CC-BY 2.0), which permits
both private and commercial use (http://creativecommons.org/licenses/by/2.0/).

Condition: For the free use of the YAML framework, a backlink to the YAML homepage (http://www.yaml.de) in a suitable place (e.g.: footer of the website or in the imprint) is required.

===================================

Notice for Tartarus

===================================

This project bundles tartarus under the MIT License.

Original source repository: https://github.com/sergiooramas/tartarus

Copyright (c) 2017 Sergio Oramas and Oriol Nieto

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.

===================================
Notice for joptsimple
===================================

This project bundles joptsimple under the MIT License.

Original source repository: https://github.com/jopt-simple/jopt-simple

Copyright (c) 2004-2016 Paul R. Holser, Jr.

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.

===================================
Notice for scopt
===================================

This project bundles scopt (v. 3.5.0) underr the MIT License.

Original source repository: https://github.com/scopt/scopt

scopt - Copyright (c) scopt contributors

See source files for details.

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.

===================================

Notice for HdrHistogram

===================================

This project bundles HdrHistogram (v. 2.1.9) under the BSD 2-Clause License

Original source repository: https://github.com/HdrHistogram/HdrHistogram

The code in this repository code was Written by Gil Tene, Michael Barker,
and Matt Warren, and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

For users of this code who wish to consume it under the "BSD" license
rather than under the public domain or CC0 contribution text mentioned
above, the code found under this directory is *also* provided under the
following license (commonly referred to as the BSD 2-Clause License). This
license does not detract from the above stated release of the code into
the public domain, and simply represents an additional license granted by
the Author.

** Beginning of "BSD 2-Clause License" text. **

Copyright (c) 2012, 2013, 2014, 2015, 2016 Gil Tene
Copyright (c) 2014 Michael Barker
Copyright (c) 2014 Matt Warren
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.

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 project includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project bundles the following dependencies under
the Apache Software License 2.0

- org.apache.lucene : lucene-core version 5.5.0
- org.apache.lucene : lucene-analyzers-common version 5.5.0
- org.apache.lucene : lucene-backward-codecs version 5.5.0
- org.apache.lucene : lucene-grouping version 5.5.0
- org.apache.lucene : lucene-highlighter version 5.5.0
- org.apache.lucene : lucene-join version 5.5.0
- org.apache.lucene : lucene-memory version 5.5.0
- org.apache.lucene : lucene-misc version 5.5.0
- org.apache.lucene : lucene-queries version 5.5.0
- org.apache.lucene : lucene-queryparser version 5.5.0
- org.apache.lucene : lucene-sandbox version 5.5.0
- org.apache.lucene : lucene-spatial version 5.5.0
- org.apache.lucene : lucene-spatial-extras version 5.5.0
- org.apache.lucene : lucene-spatial3d version 5.5.0
- org.apache.lucene : lucene-suggest version 5.5.0
- com.carrotsearch : hppc version 0.7.1
- com.google.guava : guava version 18.0
- com.google.code.findbugs : jsr305 version 1.3.9
- com.ning : compress-lzf version 1.0.2
- com.spatial4j : spatial4j version 0.5
- com.twitter : chill-java version 0.7.4
- com.fasterxml.jackson.core : jackson-core version 2.6.6
- com.fasterxml.jackson.dataformat : jackson-dataformat-smile version 2.6.6
- com.fasterxml.jackson.dataformat : jackson-dataformat-yaml version 2.6.6

- com.fasterxml.jackson.dataformat : jackson-dataformat-cbor version 2.6.6
- com.tdunning : t-digest version 3.0
- io.netty : netty version 3.10.5.Final
- joda-time : joda-time version 2.5

===================================
Notice for Yaml
===================================

This project bundles yaml (v. 1.15) under the Creative Commons License (CC-BY 2.0).

Original project website: http://www.yaml.de

Copyright (c) 2005-2013, Dirk Jesse

YAML under Creative Commons License (CC-BY 2.0)
===

The YAML framework is published under the Creative Commons Attribution 2.0 License (CC-BY 2.0), which permits
both private and commercial use (http://creativecommons.org/licenses/by/2.0/).

Condition: For the free use of the YAML framework, a backlink to the YAML homepage (http://www.yaml.de) in a suitable place (e.g.: footer of the website or in the imprint) is required.

===================================
Notice for Tartarus
===================================

This project bundles tartarus under the MIT License.

Original source repository: https://github.com/sergiooramas/tartarus

Copyright (c) 2017 Sergio Oramas and Oriol Nieto

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.

====================================
    Notice for HdrHistogram
====================================

This project bundles HdrHistogram (v. 2.1.9) under the BSD 2-Clause License

Original source repository: https://github.com/HdrHistogram/HdrHistogram

The code in this repository code was Written by Gil Tene, Michael Barker,
and Matt Warren, and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

For users of this code who wish to consume it under the "BSD" license
rather than under the public domain or CC0 contribution text mentioned
above, the code found under this directory is *also* provided under the
following license (commonly referred to as the BSD 2-Clause License). This
license does not detract from the above stated release of the code into
the public domain, and simply represents an additional license granted by
the Author.

** Beginning of "BSD 2-Clause License" text. **

Copyright (c) 2012, 2013, 2014, 2015, 2016 Gil Tene
Copyright (c) 2014 Michael Barker
Copyright (c) 2014 Matt Warren
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.

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.
All image files in the folder and its subfolders are
licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,
software distributed under the License is distributed on an
"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
KIND, either express or implied. See the License for the
specific language governing permissions and limitations
under the License.

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements.  See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License.  You may obtain a copy of the License at
*
*    http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package org.apache.flink.streaming.api.collector.selector;

import org.apache.flink.api.java.tuple.Tuple2;
import org.apache.flink.streaming.api.graph.StreamEdge;
import org.apache.flink.streaming.api.operators.Output;
import org.apache.flink.streaming.runtime.streamrecord.StreamRecord;

import java.util.Iterator;
import java.util.List;
import java.util.Set;


/**
* Special version of {@link DirectedOutput} that performs a shallow copy of the
* {@link StreamRecord} to ensure that multi-chaining works correctly.
*/
public class CopyingDirectedOutput<OUT> extends DirectedOutput<OUT> {

 @SuppressWarnings({"unchecked", "rawtypes"})
 public CopyingDirectedOutput(
   List<OutputSelector<OUT>> outputSelectors,
   List<Tuple2<Output<StreamRecord<OUT>>, StreamEdge>> outputs) {
  super(outputSelectors, outputs);
 }

 @Override
 public void collect(StreamRecord<OUT> record) {
  Set<Output<StreamRecord<OUT>>> selectedOutputs = selectOutputs(record);

  if (selectedOutputs.isEmpty()) {
   return;
  }

  Iterator<Output<StreamRecord<OUT>>> it = selectedOutputs.iterator();

  while (true) {
   Output<StreamRecord<OUT>> out = it.next();
   if (it.hasNext()) {
    // we don't have the last output
    // perform a shallow copy
    StreamRecord<OUT> shallowCopy = record.copy(record.getValue());
    out.collect(shallowCopy);
   } else {
    // this is the last output
    out.collect(record);
    break;
   }
  }
 }
}
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==

Apache Flink subcomponents:

The Apache Flink project contains subcomponents in the source code release with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of their respective licenses.

The MIT License

The Apache Flink project bundles the following files under the MIT License:

- Angular.JS v1.4.8 (http://angularjs.org) - Copyright (c) 2010-2015 Google, Inc.
- Angular Moment v0.10.3 (http://www.momentjs.com) - Copyright (C) 2013, 2014, 2015, Uri Shaked <uri@urish.org>
- Angular UI Router v0.2.15 (http://angular-ui.github.io) - Copyright (c) 2013-2015 The AngularUI Team, Karsten Sperling
- angular-drag-and-drop-lists v1.4.0 (https://github.com/marceljuenemann/angular-drag-and-drop-lists) Copyright (c) 2014 Marcel Juenemann <marcel@juenemann.cc>, Copyright (c) 2014-2016 Google Inc.
- Bootstrap v3.3.6 (http://getbootstrap.com) - Copyright (c) 2011-2015 Twitter, Inc
- d3 timeline (https://github.com/jiahuang/d3-timeline) - Copyright (c) 2012-2015 Jia Huang
- dagre v0.7.4 (https://github.com/cpettitt/dagre) - Copyright (c) 2012-2014 Chris Pettitt
- dagre-d3 v0.4.17 (https://github.com/cpettitt/dagre-d3) - Copyright (c) 2013 Chris Pettitt
- EvEmitter v1.0.2 (https://github.com/metafizzy/ev-emitter) - Copyright (C) 2016 David DeSandro
- Font Awesome (code) v4.5.0, v4.6.3 (http://fontawesome.io) - Copyright (c) 2014 Dave Gandy
- graphlib v1.0.7 (https://github.com/cpettitt/graphlib) - Copyright (c) 2012-2014 Chris Pettitt
- imagesloaded v4.1.0 (https://github.com/desandro/imagesloaded) - Copyright (C) 2016 David DeSandro
- JQuery v2.2.0 (http://jquery.com/) - Copyright 2014 jQuery Foundation and other contributors
- lodash v3.10.1 (http://dojofoundation.org) - Copyright 2012-2015 The Dojo Foundation
- moment.js v2.10.6 (http://momentjs.com/docs/) - Copyright (c) 2011-2014 Tim Wood, Iskren Chernev, Moment.js contributors
- moment-duration-format v1.3.0 (https://github.com/jsmreese/moment-duration-format) - Copyright (c) 2013 John Madhavan-Reese
- qTip2 v2.2.1 (http://qtip2.com) - Copyright (c) 2012 Craig Michael Thompson
- Split.js v1.0.6 (https://nathancahill.github.io/Split.js/) - Copyright (c) 2015 Nathan Cahill

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

BSD-style Licenses

The Apache Flink project bundles the following files under BSD licenses:

(3-clause BSD license)
- D3 v3.5.12 (http://d3js.org/) - Copyright (c) 2010-2015, Michael Bostock

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. Neither the name of the copyright holder 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.

The Open Font License

The Apache Flink project bundles the following fonts under the
Open Font License (OFT) - http://scripts.sil.org/OFL

- Font Awesome (http://fortawesome.github.io/Font-Awesome/) - Created by Dave Gandy
  -> fonts in "flink-runtime-web/web-dashboard/web/fonts"
  -> fonts in "docs/page/font-awesome/fonts"

The ISC License

The Apache Flink project contains or reuses code that is licensed under the ISC license
from the following projects:

- simplejmx (http://256stuff.com/sources/simplejmx/) Copyright (c) - Gray Watson

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL,
DIRECT, 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.

This project includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project bundles the following dependencies under
the Apache Software License 2.0

- com.facebook.presto : presto-hive version 0.185
- com.facebook.presto.hadoop : hadoop-apache2 version 2.7.3-1
- com.amazonaws : aws-java-sdk-core version 1.11.165
- com.amazonaws : aws-java-sdk-s3 version 1.11.165
- com.amazonaws : aws-java-sdk-kms version 1.11.165
- com.amazonaws : jmespath-java version 1.11.165
- software.amazon.ion : ion-java version 1.0.2
- io.airlift : stats version 0.148
- io.airlift : log version 0.148
- io.airlift : configuration version 0.148
- io.airlift : slice version 0.31
- io.airlift : units version 1.0
- com.google.guava : guava version 21.0
- com.google.code.findbugs : annotations version 2.0.3
- org.weakref : jmxutils version 1.19
- joda-time : joda-time version 2.5
- commons-logging : commons-logging version 1.1.3
- org.apache.httpcomponents : httpclient version 4.5.3
- org.apache.httpcomponents : httpcore version 4.4.6
- commons-codec : commons-codec version 1.10
- com.fasterxml.jackson.core : jackson-core version 2.7.4
- com.fasterxml.jackson.core : jackson-databind version 2.7.4
- com.fasterxml.jackson.core : jackson-annotations version 2.7.4
- com.fasterxml.jackson.dataformat : jackson-dataformat-cbor version 2.6.7

===================================
Notice for HdrHistogram
===================================

This project bundles HdrHistogram (v. 2.1.9) under the BSD 2-Clause License

Original source repository: https://github.com/HdrHistogram/HdrHistogram

The code in this repository code was Written by Gil Tene, Michael Barker, and Matt Warren, and released to the public domain, as explained at http://creativecommons.org/publicdomain/zero/1.0/

For users of this code who wish to consume it under the "BSD" license rather than under the public domain or CC0 contribution text mentioned above, the code found under this directory is *also* provided under the following license (commonly referred to as the BSD 2-Clause License). This license does not detract from the above stated release of the code into the public domain, and simply represents an additional license granted by the Author.

** Beginning of "BSD 2-Clause License" text. **

Copyright (c) 2012, 2013, 2014, 2015, 2016 Gil Tene
Copyright (c) 2014 Michael Barker
Copyright (c) 2014 Matt Warren
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.

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 project includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project bundles the following dependencies under
the Apache Software License 2.0

- commons-collections : commons-collections version 3.2.2
- commons-cli : commons-cli version 1.3.1
- commons-codec : commons-codec version 1.10
- commons-io : commons-io version 2.4
- commons-net : commons-net version 3.1
- commons-logging : commons-logging version 1.1.3
- commons-lang : commons-lang version 2.6
- commons-configuration : commons-configuration version 1.7
- commons-digester : commons-digester version 1.8.1
- commons-beanutils : commons-beanutils version 1.8.3
- org.apache.commons : commons-compress version 1.4.1
- org.apache.commons : commons-math3 version 3.5
- org.apache.httpcomponents : httpclient version 4.5.3
- org.apache.httpcomponents : httpcore version 4.4.6

- org.apache.htrace : htrace-core4 version 4.0.1-incubating
- org.apache.hadoop : hadoop-client version 2.8.1
- org.apache.hadoop : hadoop-common version 2.8.1
- org.apache.hadoop : hadoop-auth version 2.8.1
- org.apache.hadoop : hadoop-hdfs version 2.8.1
- org.apache.hadoop : hadoop-hdfs-client version 2.8.1
- org.apache.hadoop : hadoop-annotations version 2.8.1
- org.apache.hadoop : hadoop-aws version 2.8.1
- com.google.guava : guava version 11.0.2
- com.google.protobuf : protobuf-java version 2.5.0
- com.google.code.gson : gson version 2.2.4
- org.mortbay.jetty : jetty-sslengine version 6.1.26
- com.nimbusds : nimbus-jose-jwt version 3.9
- net.minidev : json-smart version 1.1.1
- joda-time : joda-time version 2.5
- com.squareup.okhttp : okhttp version 2.4.0
- com.squareup.okio : okio version 1.4.0
- com.amazonaws : aws-java-sdk-s3 version 1.10.6
- com.amazonaws : aws-java-sdk-kms version 1.10.6
- com.amazonaws : aws-java-sdk-core version 1.10.6
- org.codehaus.jackson : jackson-core-asl version 1.9.13
- org.codehaus.jackson : jackson-mapper-asl version 1.9.13
- com.fasterxml.jackson.core : jackson-core version 2.7.4
- com.fasterxml.jackson.core : jackson-databind version 2.7.4
- com.fasterxml.jackson.core : jackson-annotations version 2.7.4

This project bundles the following dependencies under
the BSD License

- xmlenc : xmlenc version 0.52 - Copyright 2003-2005, Ernst de Haan <wfe.dehaan@gmail.com>

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. Neither the name of the copyright holder 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 HOLDER 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.

Notice for org.tukaani : xz

This Java implementation of XZ has been put into the public domain, thus you can do whatever you want with it. All the files in the package have been written by Lasse Collin, but some files are heavily based on public domain code written by Igor Pavlov.

Notice for net.jcip : jcip-annotations

This project bundles net.jcip : jcip-annotations version 1.0, which has been released to the public domain.

1.25 fluent-logger-golang v1.4.0

1.25.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.26 fuzzy 0.1.0

1.26.1 Available under license :

Copyright (c) 2012 Matt York

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.

1.27 fuzzymatcher 0.1.0

1.28 fwd v1.0.0

1.28.1 Available under license :

Copyright (c) 2014-2015, Philip Hofer

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.

1.29 github.com/beorn7/perks 3a771d992973f24aa725d07868b467d1ddfceaf b :1.0

1.29.1 Available under license :

Copyright (C) 2013 Blake Mizerany

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.

1.30 go-ini/ini v1.42.0

1.30.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and

distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of

the NOTICE file are for informational purposes only and do not modify the
License. You may add Your own attribution notices within Derivative Works that
You distribute, alongside or as an addendum to the NOTICE text from the Work,
provided that such additional attribution notices cannot be construed as
modifying the License.
You may add Your own copyright statement to Your modifications and may provide
additional or different license terms and conditions for use, reproduction, or
distribution of Your modifications, or for any such Derivative Works as a whole,
provided Your use, reproduction, and distribution of the Work otherwise complies
with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted
for inclusion in the Work by You to the Licensor shall be under the terms and
conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of
any separate license agreement you may have executed with Licensor regarding
such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks,
service marks, or product names of the Licensor, except as required for
reasonable and customary use in describing the origin of the Work and
reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the
Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied,
including, without limitation, any warranties or conditions of TITLE,
NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are
solely responsible for determining the appropriateness of using or
redistributing the Work and assume any risks associated with Your exercise of
permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence),
contract, or otherwise, unless required by applicable law (such as deliberate
and grossly negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special, incidental,
or consequential damages of any character arising as a result of this License or
out of the use or inability to use the Work (including but not limited to
damages for loss of goodwill, work stoppage, computer failure or malfunction, or
any and all other commercial damages or losses), even if such Contributor has

been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2014 Unknwon

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.31 go-resiliency 1.0.0

1.31.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Evan Huus

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.

1.32 go-spew v1.1.0

1.32.1 Available under license :

ISC License

Copyright (c) 2012-2016 Dave Collins <dave@davec.name>

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" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, 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.

1.33 go-xerial-snappy master

1.33.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Evan Huus

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.

1.34 golang-snappy v0.0.1

1.34.1 Available under license :

Copyright (c) 2011 The Snappy-Go 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 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.
# This is the official list of people who can contribute
# (and typically have contributed) code to the Snappy-Go repository.

```
# The AUTHORS file lists the copyright holders; this file
# lists people.  For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# The submission process automatically checks to make sure
# that people submitting code are listed in this file (by email address).
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
#     http://code.google.com/legal/individual-cla-v1.0.html
#     http://code.google.com/legal/corporate-cla-v1.0.html
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate CLA was used.

# Names should be added to this file like so:
#     Name <email address>

# Please keep the list sorted.

Damian Gryski <dgryski@gmail.com>
Jan Mercl <0xjnml@gmail.com>
Kai Backman <kaib@golang.org>
Marc-Antoine Ruel <maruel@chromium.org>
Nigel Tao <nigeltao@golang.org>
Rob Pike <r@golang.org>
Rodolfo Carvalho <rhcarvalho@gmail.com>
Russ Cox <rsc@golang.org>
Sebastien Binet <seb.binet@gmail.com>
```

1.35 golang/genproto master

1.35.1 Available under license :

```
                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.
```

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.
/*
* Copyright 2002-2014 the original author or authors.
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

1.36 golang/grpc v1.21.0-dev

1.36.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.37 golang/x/crypto master

1.37.1 Available under license :

Copyright (c) 2009 The Go 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 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.
# This source code was written by the Go contributors.
# The master list of contributors is in the main Go distribution,
# visible at https://tip.golang.org/CONTRIBUTORS.

1.38 golang/x/net master

1.38.1 Available under license :

Copyright (c) 2009 The Go 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 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.
# This source code was written by the Go contributors.
# The master list of contributors is in the main Go distribution,
# visible at http://tip.golang.org/CONTRIBUTORS.

1.39 golang/x/sys master

1.39.1 Available under license :

Copyright (c) 2009 The Go 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 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.
# This source code was written by the Go contributors.
# The master list of contributors is in the main Go distribution,
# visible at http://tip.golang.org/CONTRIBUTORS.

1.40 golang/x/text master

1.40.1 Available under license :

Copyright (c) 2009 The Go 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 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.
# This source code was written by the Go contributors.
# The master list of contributors is in the main Go distribution,
# visible at http://tip.golang.org/CONTRIBUTORS.

1.41 golang_protobuf_extensions v1.0.1

1.41.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly

negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

/*
* Copyright 2002-2014 the original author or authors.

*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   https://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

1.42 gorilla-mux v1.7.0

1.42.1 Available under license :

Copyright (c) 2012 Rodrigo Moraes. 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.

1.43 gorilla-websocket v1.4.0

1.43.1 Available under license :

Copyright (c) 2013 The Gorilla WebSocket 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.

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.

1.44 Guava 18.0

1.44.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following

conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may

choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
  Copyright [yyyy] [name of copyright owner]

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
```

```
/*
* Copyright (C) 2010 The Guava Authors
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

1.45 httpclient 4.5.6

1.45.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache HttpClient
Copyright 1999-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

1.46 influxdb v1.2.0

1.46.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013-2016 Errplane Inc.

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.
# List
- bootstrap 3.3.5 [MIT LICENSE](https://github.com/twbs/bootstrap/blob/master/LICENSE)
- collectd.org [ISC LICENSE](https://github.com/collectd/go-collectd/blob/master/LICENSE)

- github.com/BurntSushi/toml [WTFPL LICENSE](https://github.com/BurntSushi/toml/blob/master/COPYING)
- github.com/bmizerany/pat [MIT LICENSE](https://github.com/bmizerany/pat#license)
- github.com/boltdb/bolt [MIT LICENSE](https://github.com/boltdb/bolt/blob/master/LICENSE)
- github.com/cespare/xxhash [MIT LICENSE](https://github.com/cespare/xxhash/blob/master/LICENSE.txt)
- github.com/davecgh/go-spew/spew [ISC LICENSE](https://github.com/davecgh/go-spew/blob/master/LICENSE)
- github.com/dgrijalva/jwt-go [MIT LICENSE](https://github.com/dgrijalva/jwt-go/blob/master/LICENSE)
- github.com/dgryski/go-bits [MIT LICENSE](https://github.com/dgryski/go-bits/blob/master/LICENSE)
- github.com/dgryski/go-bitstream [MIT LICENSE](https://github.com/dgryski/go-bitstream/blob/master/LICENSE)
- github.com/gogo/protobuf/proto [BSD LICENSE](https://github.com/gogo/protobuf/blob/master/LICENSE)
- github.com/golang/snappy [BSD LICENSE](https://github.com/golang/snappy/blob/master/LICENSE)
- github.com/influxdata/usage-client [MIT LICENSE](https://github.com/influxdata/usage-client/blob/master/LICENSE.txt)
- github.com/jwilder/encoding [MIT LICENSE](https://github.com/jwilder/encoding/blob/master/LICENSE)
- github.com/paulbellamy/ratecounter [MIT LICENSE](https://github.com/paulbellamy/ratecounter/blob/master/LICENSE)
- github.com/peterh/liner [MIT LICENSE](https://github.com/peterh/liner/blob/master/COPYING)
- github.com/rakyll/statik [APACHE LICENSE](https://github.com/rakyll/statik/blob/master/LICENSE)
- github.com/retailnext/hllpp [BSD LICENSE](https://github.com/retailnext/hllpp/blob/master/LICENSE)
- github.com/uber-go/atomic [MIT LICENSE](https://github.com/uber-go/atomic/blob/master/LICENSE.txt)
- github.com/uber-go/zap [MIT LICENSE](https://github.com/uber-go/zap/blob/master/LICENSE.txt)
- glyphicons [LICENSE](http://glyphicons.com/license/)
- golang.org/x/crypto [BSD LICENSE](https://github.com/golang/crypto/blob/master/LICENSE)
- jquery 2.1.4 [MIT LICENSE](https://github.com/jquery/jquery/blob/master/LICENSE.txt)
- react 0.13.3 [BSD LICENSE](https://github.com/facebook/react/blob/master/LICENSE)

1.47 jackson-core 2.9.4

1.47.1 Available under license :

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.48 jackson-core 2.7.5

1.48.1 Available under license :

This copy of Jackson JSON processor streaming parser/generator is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
# Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

## Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

## Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or

conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.49 js-cookie 2.2.0

1.49.1 Available under license :

Copyright 2014 Klaus Hartl

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.

1.50 json-cpp 1.8.4

1.50.1 Available under license :

The JsonCpp library's source code, including accompanying documentation,
tests and demonstration applications, are licensed under the following
conditions...

Baptiste Lepilleur and The JsonCpp Authors explicitly disclaim 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
The JsonCpp Authors, 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 and The JsonCpp Authors

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.

The JsonCpp library's source code, including accompanying documentation,
tests and demonstration applications, are licensed under the following
conditions...

Baptiste Lepilleur and The JsonCpp Authors explicitly disclaim 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
The JsonCpp Authors, 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 and The JsonCpp Authors

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.

1.51 json-path 2.3.0

1.51.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2017 Jayway

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.52 json-schema-faker 0.5.0-rc9

1.52.1 Available under license :

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.

1.53 kafka-python 1.4.6

1.53.1 Available under license :

Metadata-Version: 1.1
Name: kafka-python
Version: 1.4.6
Summary: Pure Python client for Apache Kafka
Home-page: https://github.com/dpkp/kafka-python
Author: Dana Powers
Author-email: dana.powers@gmail.com
License: Apache License 2.0
Description: Kafka Python client

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 2015 David Arthur

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.54 klauspost/crc32 v1.2.0

1.54.1 Available under license :

Copyright (c) 2012 The Go Authors. All rights reserved.
Copyright (c) 2015 Klaus Post

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.

1.55 libcurl 7.61.1

1.55.1 Available under license :

License Mixing
==============

libcurl can be built to use a fair amount of various third party libraries, libraries that are written and provided by other parties that are distributed using their own licenses. Even libcurl itself contains code that may cause problems to some. This document attempts to describe what licenses libcurl and the other libraries use and what possible dilemmas linking and mixing them all can lead to for end users.

I am not a lawyer and this is not legal advice!

One common dilemma is that [GPL](https://www.gnu.org/licenses/gpl.html) licensed code is not allowed to be linked with code licensed under the [Original BSD license](https://spdx.org/licenses/BSD-4-Clause.html) (with the announcement clause). You may still build your own copies that use them all, but distributing them as binaries would be to violate the GPL license - unless you accompany your license with an [exception](https://www.gnu.org/licenses/gpl-faq.html#GPLIncompatibleLibs). This particular problem was addressed when the [Modified BSD license](https://opensource.org/licenses/BSD-3-Clause) was created, which does not have the announcement clause that collides with GPL.

## libcurl

Uses an [MIT style license](https://curl.haxx.se/docs/copyright.html) that is very liberal.

## OpenSSL

(May be used for SSL/TLS support) Uses an Original BSD-style license with an announcement clause that makes it "incompatible" with GPL. You are not allowed to ship binaries that link with OpenSSL that includes GPL code (unless that specific GPL code includes an exception for OpenSSL - a habit that is growing more and more common). If OpenSSL's licensing is a problem for you, consider using another TLS library.

## GnuTLS

(May be used for SSL/TLS support) Uses the [LGPL](https://www.gnu.org/licenses/lgpl.html) license. If this is a problem for you, consider using another TLS library. Also note that GnuTLS itself depends on and uses other libs (libgcrypt and libgpg-error) and they too are LGPL- or GPL-licensed.

## WolfSSL

(May be used for SSL/TLS support) Uses the GPL license or a proprietary license. If this is a problem for you, consider using another TLS library.

## NSS

(May be used for SSL/TLS support) Is covered by the [MPL](https://www.mozilla.org/MPL/) license, the GPL license and the LGPL license. You may choose to license the code under MPL terms, GPL terms, or LGPL terms. These licenses grant you different permissions and impose different obligations. You should select the license that best meets your needs.

## axTLS

(May be used for SSL/TLS support) Uses a Modified BSD-style license.

## mbedTLS

(May be used for SSL/TLS support) Uses the [Apache 2.0 license](https://opensource.org/licenses/Apache-2.0) or the GPL license. You may choose to license the code under Apache 2.0 terms or GPL terms. These licenses grant you different permissions and impose different obligations. You should select the license that best meets your needs.

## BoringSSL

(May be used for SSL/TLS support) As an OpenSSL fork, it has the same license as that.

## libressl

(May be used for SSL/TLS support) As an OpenSSL fork, it has the same license as that.

## c-ares

(Used for asynchronous name resolves) Uses an MIT license that is very liberal and imposes no restrictions on any other library or part you may link with.

## zlib

(Used for compressed Transfer-Encoding support) Uses an MIT-style license that shouldn't collide with any other library.

## MIT Kerberos

(May be used for GSS support) MIT licensed, that shouldn't collide with any other parts.

## Heimdal

(May be used for GSS support) Heimdal is Original BSD licensed with the announcement clause.

## GNU GSS

(May be used for GSS support) GNU GSS is GPL licensed. Note that you may not distribute binary curl packages that uses this if you build curl to also link and use any Original BSD licensed libraries!

## libidn

(Used for IDNA support) Uses the GNU Lesser General Public License [3]. LGPL is a variation of GPL with slightly less aggressive "copyleft". This license requires more requirements to be met when distributing binaries, see the license for details. Also note that if you distribute a binary that includes this library, you must also include the full LGPL license text. Please properly point out what parts of the distributed package that the license addresses.

## OpenLDAP

(Used for LDAP support) Uses a Modified BSD-style license. Since libcurl uses OpenLDAP as a shared library only, I have not heard of anyone that ships OpenLDAP linked with libcurl in an app.

## libssh2

(Used for scp and sftp support) libssh2 uses a Modified BSD-style license.
COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2018, 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.

```
# libtool.m4 - Configure libtool for the host system. -*-Autoconf-*-
#
#   Copyright (C) 1996-2001, 2003-2015 Free Software Foundation, Inc.
#   Written by Gordon Matzigkeit, 1996
#
# This file is free software; the Free Software Foundation gives
# unlimited permission to copy and/or distribute it, with or without
# modifications, as long as this notice is preserved.

m4_define([_LT_COPYING], [dnl
# Copyright (C) 2014 Free Software Foundation, Inc.
# This is free software; see the source for copying conditions.  There is NO
# warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

# GNU Libtool 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 of the License, or
# (at your option) any later version.
#
# As a special exception to the GNU General Public License, if you
# distribute this file as part of a program or library that is built
# using GNU Libtool, you may include this file under the  same
# distribution terms that you use for the rest of that program.
#
# GNU Libtool 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, see <http://www.gnu.org/licenses/>.
])

# serial 58 LT_INIT


# LT_PREREQ(VERSION)
# ------------------
# Complain and exit if this libtool version is less that VERSION.
m4_defun([LT_PREREQ],
[m4_if(m4_version_compare(m4_defn([LT_PACKAGE_VERSION]), [$1]), -1,
       [m4_default([$3],
      [m4_fatal([Libtool version $1 or higher is required],
                63)])],
       [$2])])


# _LT_CHECK_BUILDDIR
# ------------------
# Complain if the absolute build directory name contains unusual characters
m4_defun([_LT_CHECK_BUILDDIR],
[case `pwd` in
 *\ * | *\ *)
   AC_MSG_WARN([Libtool does not cope well with whitespace in `pwd`]) ;;
esac
])


# LT_INIT([OPTIONS])
# ------------------
AC_DEFUN([LT_INIT],
[AC_PREREQ([2.62])dnl We use AC_PATH_PROGS_FEATURE_CHECK
AC_REQUIRE([AC_CONFIG_AUX_DIR_DEFAULT])dnl
AC_BEFORE([$0], [LT_LANG])dnl
AC_BEFORE([$0], [LT_OUTPUT])dnl
AC_BEFORE([$0], [LTDL_INIT])dnl
m4_require([_LT_CHECK_BUILDDIR])dnl

dnl Autoconf doesn't catch unexpanded LT_ macros by default:
m4_pattern_forbid([^_?LT_[A-Z_]+$])dnl
m4_pattern_allow([^(_LT_EOF|LT_DLGLOBAL|LT_DLLAZY_OR_NOW|LT_MULTI_MODULE)$])dnl
dnl aclocal doesn't pull ltoptions.m4, ltsugar.m4, or ltversion.m4
dnl unless we require an AC_DEFUNed macro:
```

```
AC_REQUIRE([LTOPTIONS_VERSION])dnl
AC_REQUIRE([LTSUGAR_VERSION])dnl
AC_REQUIRE([LTVERSION_VERSION])dnl
AC_REQUIRE([LTOBSOLETE_VERSION])dnl
m4_require([_LT_PROG_LTMAIN])dnl

_LT_SHELL_INIT([SHELL=${CONFIG_SHELL-/bin/sh}])

dnl Parse OPTIONS
_LT_SET_OPTIONS([$0], [$1])

# This can be used to rebuild libtool when needed
LIBTOOL_DEPS=$ltmain

# Always use our own libtool.
LIBTOOL='$(SHELL) $(top_builddir)/libtool'
AC_SUBST(LIBTOOL)dnl

_LT_SETUP

# Only expand once:
m4_define([LT_INIT])
])# LT_INIT

# Old names:
AU_ALIAS([AC_PROG_LIBTOOL], [LT_INIT])
AU_ALIAS([AM_PROG_LIBTOOL], [LT_INIT])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AC_PROG_LIBTOOL], [])
dnl AC_DEFUN([AM_PROG_LIBTOOL], [])


# _LT_PREPARE_CC_BASENAME
# -----------------------
m4_defun([_LT_PREPARE_CC_BASENAME], [
# Calculate cc_basename.  Skip known compiler wrappers and cross-prefix.
func_cc_basename ()
{
   for cc_temp in @S|@*""; do
     case $cc_temp in
       compile | *[[\\/]]compile | ccache | *[[\\/]]ccache ) ;;
       distcc | *[[\\/]]distcc | purify | *[[\\/]]purify ) ;;
       \-*) ;;
       *) break;;
     esac
   done
   func_cc_basename_result=`$ECHO "$cc_temp" | $SED "s%.*/%%; s%^$host_alias-%%"`
}
```

```
])# _LT_PREPARE_CC_BASENAME


# _LT_CC_BASENAME(CC)
# -------------------
# It would be clearer to call AC_REQUIREs from _LT_PREPARE_CC_BASENAME,
# but that macro is also expanded into generated libtool script, which
# arranges for $SED and $ECHO to be set by different means.
m4_defun([_LT_CC_BASENAME],
[m4_require([_LT_PREPARE_CC_BASENAME])dnl
AC_REQUIRE([_LT_DECL_SED])dnl
AC_REQUIRE([_LT_PROG_ECHO_BACKSLASH])dnl
func_cc_basename $1
cc_basename=$func_cc_basename_result
])


# _LT_FILEUTILS_DEFAULTS
# ----------------------
# It is okay to use these file commands and assume they have been set
# sensibly after 'm4_require([_LT_FILEUTILS_DEFAULTS])'.
m4_defun([_LT_FILEUTILS_DEFAULTS],
[: ${CP="cp -f"}
: ${MV="mv -f"}
: ${RM="rm -f"}
])# _LT_FILEUTILS_DEFAULTS


# _LT_SETUP
# ---------
m4_defun([_LT_SETUP],
[AC_REQUIRE([AC_CANONICAL_HOST])dnl
AC_REQUIRE([AC_CANONICAL_BUILD])dnl
AC_REQUIRE([_LT_PREPARE_SED_QUOTE_VARS])dnl
AC_REQUIRE([_LT_PROG_ECHO_BACKSLASH])dnl

_LT_DECL([], [PATH_SEPARATOR], [1], [The PATH separator for the build system])dnl
dnl
_LT_DECL([], [host_alias], [0], [The host system])dnl
_LT_DECL([], [host], [0])dnl
_LT_DECL([], [host_os], [0])dnl
dnl
_LT_DECL([], [build_alias], [0], [The build system])dnl
_LT_DECL([], [build], [0])dnl
_LT_DECL([], [build_os], [0])dnl
dnl
AC_REQUIRE([AC_PROG_CC])dnl
AC_REQUIRE([LT_PATH_LD])dnl
AC_REQUIRE([LT_PATH_NM])dnl
dnl
AC_REQUIRE([AC_PROG_LN_S])dnl
test -z "$LN_S" && LN_S="ln -s"
_LT_DECL([], [LN_S], [1], [Whether we need soft or hard links])dnl
dnl
AC_REQUIRE([LT_CMD_MAX_LEN])dnl
_LT_DECL([objext], [ac_objext], [0], [Object file suffix (normally "o")])dnl
_LT_DECL([], [exeext], [0], [Executable file suffix (normally "")])dnl
dnl
m4_require([_LT_FILEUTILS_DEFAULTS])dnl
m4_require([_LT_CHECK_SHELL_FEATURES])dnl
m4_require([_LT_PATH_CONVERSION_FUNCTIONS])dnl
m4_require([_LT_CMD_RELOAD])dnl
m4_require([_LT_CHECK_MAGIC_METHOD])dnl
m4_require([_LT_CHECK_SHAREDLIB_FROM_LINKLIB])dnl
m4_require([_LT_CMD_OLD_ARCHIVE])dnl
m4_require([_LT_CMD_GLOBAL_SYMBOLS])dnl
m4_require([_LT_WITH_SYSROOT])dnl
m4_require([_LT_CMD_TRUNCATE])dnl

_LT_CONFIG_LIBTOOL_INIT([
# See if we are running on zsh, and set the options that allow our
# commands through without removal of \ escapes INIT.
if test -n "\${ZSH_VERSION+set}"; then
  setopt NO_GLOB_SUBST
fi
])
if test -n "${ZSH_VERSION+set}"; then
  setopt NO_GLOB_SUBST
fi

_LT_CHECK_OBJDIR

m4_require([_LT_TAG_COMPILER])dnl

case $host_os in
aix3*)
 # AIX sometimes has problems with the GCC collect2 program.  For some
 # reason, if we set the COLLECT_NAMES environment variable, the problems
 # vanish in a puff of smoke.
 if test set != "${COLLECT_NAMES+set}"; then
   COLLECT_NAMES=
   export COLLECT_NAMES
 fi
 ;;
esac
```

```
# Global variables:
ofile=libtool
can_build_shared=yes

# All known linkers require a '.a' archive for static linking (except MSVC,
# which needs '.lib').
libext=a

with_gnu_ld=$lt_cv_prog_gnu_ld

old_CC=$CC
old_CFLAGS=$CFLAGS

# Set sane defaults for various variables
test -z "$CC" && CC=cc
test -z "$LTCC" && LTCC=$CC
test -z "$LTCFLAGS" && LTCFLAGS=$CFLAGS
test -z "$LD" && LD=ld
test -z "$ac_objext" && ac_objext=o

_LT_CC_BASENAME([$compiler])

# Only perform the check for file, if the check method requires it
test -z "$MAGIC_CMD" && MAGIC_CMD=file
case $deplibs_check_method in
file_magic*)
 if test "$file_magic_cmd" = '$MAGIC_CMD'; then
   _LT_PATH_MAGIC
 fi
 ;;
esac

# Use C for the default configuration in the libtool script
LT_SUPPORTED_TAG([CC])
_LT_LANG_C_CONFIG
_LT_LANG_DEFAULT_CONFIG
_LT_CONFIG_COMMANDS
])# _LT_SETUP


# _LT_PREPARE_SED_QUOTE_VARS
# --------------------------
# Define a few sed substitution that help us do robust quoting.
m4_defun([_LT_PREPARE_SED_QUOTE_VARS],
[# Backslashify metacharacters that are still active within
# double-quoted strings.
sed_quote_subst='s/\([["`$\\]]\)/\\\1/g'
```

```
# Same as above, but do not quote variable references.
double_quote_subst='s/\([["`\\]]\)/\\\1/g'

# Sed substitution to delay expansion of an escaped shell variable in a
# double_quote_subst'ed string.
delay_variable_subst='s/\\\\\\\\\\\$/\\\\\\$/g'

# Sed substitution to delay expansion of an escaped single quote.
delay_single_quote_subst='s/'\''/'\'\\\\\\\'\''/g'

# Sed substitution to avoid accidental globbing in evaled expressions
no_glob_subst='s/\*/\\\*/g'
])

# _LT_PROG_LTMAIN
# ---------------
# Note that this code is called both from 'configure', and 'config.status'
# now that we use AC_CONFIG_COMMANDS to generate libtool.  Notably,
# 'config.status' has no value for ac_aux_dir unless we are using Automake,
# so we pass a copy along to make sure it has a sensible value anyway.
m4_defun([_LT_PROG_LTMAIN],
[m4_ifdef([AC_REQUIRE_AUX_FILE], [AC_REQUIRE_AUX_FILE([ltmain.sh])])dnl
_LT_CONFIG_LIBTOOL_INIT([ac_aux_dir='$ac_aux_dir'])
ltmain=$ac_aux_dir/ltmain.sh
])# _LT_PROG_LTMAIN



## ------------------------------------- ##
## Accumulate code for creating libtool. ##
## ------------------------------------- ##

# So that we can recreate a full libtool script including additional
# tags, we accumulate the chunks of code to send to AC_CONFIG_COMMANDS
# in macros and then make a single call at the end using the 'libtool'
# label.


# _LT_CONFIG_LIBTOOL_INIT([INIT-COMMANDS])
# ----------------------------------------
# Register INIT-COMMANDS to be passed to AC_CONFIG_COMMANDS later.
m4_define([_LT_CONFIG_LIBTOOL_INIT],
[m4_ifval([$1],
          [m4_append([_LT_OUTPUT_LIBTOOL_INIT],
                     [$1
])])])

# Initialize.
m4_define([_LT_OUTPUT_LIBTOOL_INIT])
```

```
# _LT_CONFIG_LIBTOOL([COMMANDS])
# ------------------------------
# Register COMMANDS to be passed to AC_CONFIG_COMMANDS later.
m4_define([_LT_CONFIG_LIBTOOL],
[m4_ifval([$1],
        [m4_append([_LT_OUTPUT_LIBTOOL_COMMANDS],
                  [$1
])])])

# Initialize.
m4_define([_LT_OUTPUT_LIBTOOL_COMMANDS])


# _LT_CONFIG_SAVE_COMMANDS([COMMANDS], [INIT_COMMANDS])
# --------------------------------------------------
m4_defun([_LT_CONFIG_SAVE_COMMANDS],
[_LT_CONFIG_LIBTOOL([$1])
_LT_CONFIG_LIBTOOL_INIT([$2])
])


# _LT_FORMAT_COMMENT([COMMENT])
# -----------------------------
# Add leading comment marks to the start of each line, and a trailing
# full-stop to the whole comment if one is not present already.
m4_define([_LT_FORMAT_COMMENT],
[m4_ifval([$1], [
m4_bpatsubst([m4_bpatsubst([$1], [^ *], [# ])],
            [['`$\]], [\\\&])]m4_bmatch([$1], [[!?.]$], [], [.])
)])



## ------------------------ ##
## FIXME: Eliminate VARNAME ##
## ------------------------ ##


# _LT_DECL([CONFIGNAME], VARNAME, VALUE, [DESCRIPTION], [IS-TAGGED?])
# ---------------------------------------------------------------
# CONFIGNAME is the name given to the value in the libtool script.
# VARNAME is the (base) name used in the configure script.
# VALUE may be 0, 1 or 2 for a computed quote escaped value based on
# VARNAME.  Any other value will be used directly.
m4_define([_LT_DECL],
[lt_if_append_uniq([lt_decl_varnames], [$2], [, ],
  [lt_dict_add_subkey([lt_decl_dict], [$2], [libtool_name],
 [m4_ifval([$1], [$1], [$2])])
  lt_dict_add_subkey([lt_decl_dict], [$2], [value], [$3])
  m4_ifval([$4],
 [lt_dict_add_subkey([lt_decl_dict], [$2], [description], [$4])])
  lt_dict_add_subkey([lt_decl_dict], [$2],
 [tagged?], [m4_ifval([$5], [yes], [no])])])
])


# _LT_TAGDECL([CONFIGNAME], VARNAME, VALUE, [DESCRIPTION])
# --------------------------------------------------------
m4_define([_LT_TAGDECL], [_LT_DECL([$1], [$2], [$3], [$4], [yes])])


# lt_decl_tag_varnames([SEPARATOR], [VARNAME1...])
# ------------------------------------------------
m4_define([lt_decl_tag_varnames],
[_lt_decl_filter([tagged?], [yes], $@)])


# _lt_decl_filter(SUBKEY, VALUE, [SEPARATOR], [VARNAME1..])
# ---------------------------------------------------------
m4_define([_lt_decl_filter],
[m4_case([$#],
 [0], [m4_fatal([$0: too few arguments: $#])],
 [1], [m4_fatal([$0: too few arguments: $#: $1])],
 [2], [lt_dict_filter([lt_decl_dict], [$1], [$2], [], lt_decl_varnames)],
 [3], [lt_dict_filter([lt_decl_dict], [$1], [$2], [$3], lt_decl_varnames)],
 [lt_dict_filter([lt_decl_dict], $@)])[]dnl
])


# lt_decl_quote_varnames([SEPARATOR], [VARNAME1...])
# --------------------------------------------------
m4_define([lt_decl_quote_varnames],
[_lt_decl_filter([value], [1], $@)])


# lt_decl_dquote_varnames([SEPARATOR], [VARNAME1...])
# ---------------------------------------------------
m4_define([lt_decl_dquote_varnames],
[_lt_decl_filter([value], [2], $@)])


# lt_decl_varnames_tagged([SEPARATOR], [VARNAME1...])
# ---------------------------------------------------
m4_define([lt_decl_varnames_tagged],
[m4_assert([$# <= 2])dnl
_$0(m4_quote(m4_default([$1], [[, ]])),
   m4_ifval([$2], [[$2]], [m4_dquote(lt_decl_tag_varnames)]),
   m4_split(m4_normalize(m4_quote(_LT_TAGS)), [ ]))])
m4_define([_lt_decl_varnames_tagged],
[m4_ifval([$3], [lt_combine([$1], [$2], [_], $3)])])


# lt_decl_all_varnames([SEPARATOR], [VARNAME1...])
# ------------------------------------------------
m4_define([lt_decl_all_varnames],
[_$0(m4_quote(m4_default([$1], [[, ]])),
    m4_if([$2], [],
    m4_quote(lt_decl_varnames),
 m4_quote(m4_shift($@))))[]dnl
])
m4_define([_lt_decl_all_varnames],
[lt_join($@, lt_decl_varnames_tagged([$1],
   lt_decl_tag_varnames([[, ]], m4_shift($@))))dnl
])


# _LT_CONFIG_STATUS_DECLARE([VARNAME])
# ------------------------------------
# Quote a variable value, and forward it to 'config.status' so that its
# declaration there will have the same value as in 'configure'.  VARNAME
# must have a single quote delimited value for this to work.
m4_define([_LT_CONFIG_STATUS_DECLARE],
[$1='`$ECHO "$][$1" | $SED "$delay_single_quote_subst"`'])


# _LT_CONFIG_STATUS_DECLARATIONS
# ------------------------------
# We delimit libtool config variables with single quotes, so when
# we write them to config.status, we have to be sure to quote all
# embedded single quotes properly.  In configure, this macro expands
# each variable declared with _LT_DECL (and _LT_TAGDECL) into:
#
#    <var>='`$ECHO "$<var>" | $SED "$delay_single_quote_subst"`'
m4_defun([_LT_CONFIG_STATUS_DECLARATIONS],
[m4_foreach([_lt_var], m4_quote(lt_decl_all_varnames),
   [m4_n([_LT_CONFIG_STATUS_DECLARE(_lt_var)])])])


# _LT_LIBTOOL_TAGS
# ----------------
# Output comment and list of tags supported by the script
m4_defun([_LT_LIBTOOL_TAGS],
[_LT_FORMAT_COMMENT([The names of the tagged configurations supported by this script])dnl
available_tags='_LT_TAGS'dnl
])


# _LT_LIBTOOL_DECLARE(VARNAME, [TAG])
# ----------------------------------
# Extract the dictionary values for VARNAME (optionally with TAG) and
# expand to a commented shell variable setting:
#
#    # Some comment about what VAR is for.
#    visible_name=$lt_internal_name
m4_define([_LT_LIBTOOL_DECLARE],
[_LT_FORMAT_COMMENT(m4_quote(lt_dict_fetch([lt_decl_dict], [$1],
        [description])))[]dnl
m4_pushdef([_libtool_name],
   m4_quote(lt_dict_fetch([lt_decl_dict], [$1], [libtool_name])))[]dnl
m4_case(m4_quote(lt_dict_fetch([lt_decl_dict], [$1], [value])),
   [0], [_libtool_name=[$]$1],
   [1], [_libtool_name=$lt_[]$1],
   [2], [_libtool_name=$lt_[]$1],
   [_libtool_name=lt_dict_fetch([lt_decl_dict], [$1], [value])])[]dnl
m4_ifval([$2], [_$2])[]m4_popdef([_libtool_name])[]dnl
])


# _LT_LIBTOOL_CONFIG_VARS
# -----------------------
# Produce commented declarations of non-tagged libtool config variables
# suitable for insertion in the LIBTOOL CONFIG section of the 'libtool'
# script.  Tagged libtool config variables (even for the LIBTOOL CONFIG
# section) are produced by _LT_LIBTOOL_TAG_VARS.
m4_defun([_LT_LIBTOOL_CONFIG_VARS],
[m4_foreach([_lt_var],
   m4_quote(_lt_decl_filter([tagged?], [no], [], lt_decl_varnames)),
   [m4_n([_LT_LIBTOOL_DECLARE(_lt_var)])])])


# _LT_LIBTOOL_TAG_VARS(TAG)
# -------------------------
m4_define([_LT_LIBTOOL_TAG_VARS],
[m4_foreach([_lt_var], m4_quote(lt_decl_tag_varnames),
   [m4_n([_LT_LIBTOOL_DECLARE(_lt_var, [$1])])])])


# _LT_TAGVAR(VARNAME, [TAGNAME])
# ------------------------------
m4_define([_LT_TAGVAR], [m4_ifval([$2], [$1_$2], [$1])])
```

```
# _LT_CONFIG_COMMANDS
# -------------------
# Send accumulated output to $CONFIG_STATUS.  Thanks to the lists of
# variables for single and double quote escaping we saved from calls
# to _LT_DECL, we can put quote escaped variables declarations
# into 'config.status', and then the shell code to quote escape them in
# for loops in 'config.status'.  Finally, any additional code accumulated
# from calls to _LT_CONFIG_LIBTOOL_INIT is expanded.
m4_defun([_LT_CONFIG_COMMANDS],
[AC_PROVIDE_IFELSE([LT_OUTPUT],
 dnl If the libtool generation code has been placed in $CONFIG_LT,
 dnl instead of duplicating it all over again into config.status,
 dnl then we will have config.status run $CONFIG_LT later, so it
 dnl needs to know what name is stored there:
       [AC_CONFIG_COMMANDS([libtool],
           [$SHELL $CONFIG_LT || AS_EXIT(1)], [CONFIG_LT='$CONFIG_LT'])],
   dnl If the libtool generation code is destined for config.status,
   dnl expand the accumulated commands and init code now:
   [AC_CONFIG_COMMANDS([libtool],
       [_LT_OUTPUT_LIBTOOL_COMMANDS], [_LT_OUTPUT_LIBTOOL_COMMANDS_INIT])])
])#_LT_CONFIG_COMMANDS


# Initialize.
m4_define([_LT_OUTPUT_LIBTOOL_COMMANDS_INIT],
[

# The HP-UX ksh and POSIX shell print the target directory to stdout
# if CDPATH is set.
(unset CDPATH) >/dev/null 2>&1 && unset CDPATH

sed_quote_subst='$sed_quote_subst'
double_quote_subst='$double_quote_subst'
delay_variable_subst='$delay_variable_subst'
_LT_CONFIG_STATUS_DECLARATIONS
LTCC='$LTCC'
LTCFLAGS='$LTCFLAGS'
compiler='$compiler_DEFAULT'

# A function that is used when there is no print builtin or printf.
func_fallback_echo ()
{
 eval 'cat <<_LTECHO_EOF
\$[]1
_LTECHO_EOF'
}
# Quote evaled strings.
for var in lt_decl_all_varnames([[ \
]], lt_decl_quote_varnames); do
  case \`eval \\\\\$ECHO \\\\""\\\\\$\$var"\\\\"\` in
  *[[\\\\\\\`\\"\\\$]]*)
    eval "lt_\$var=\\\\\\"\\\`\\\$ECHO \\"\\\$\$var\\" | \\\$SED \\"\\\$sed_quote_subst\\"\\\`\\\\\\"" ## exclude from
sc_prohibit_nested_quotes
    ;;
  *)
    eval "lt_\$var=\\\\\\"\\\$\$var\\\\\\""
    ;;
  esac
done

# Double-quote double-evaled strings.
for var in lt_decl_all_varnames([[ \
]], lt_decl_dquote_varnames); do
  case \`eval \\\\\$ECHO \\\\""\\\\\$\$var"\\\\"\` in
  *[[\\\\\\\`\\"\\\$]]*)
    eval "lt_\$var=\\\\\\"\\\`\\\$ECHO \\"\\\$\$var\\" | \\\$SED -e \\"\\\$double_quote_subst\\" -e
\\"\\\$sed_quote_subst\\" -e \\"\\\$delay_variable_subst\\"\\\`\\\\\\"" ## exclude from sc_prohibit_nested_quotes
    ;;
  *)
    eval "lt_\$var=\\\\\\"\\\$\$var\\\\\\""
    ;;
  esac
done

_LT_OUTPUT_LIBTOOL_INIT
])

# _LT_GENERATED_FILE_INIT(FILE, [COMMENT])
# ------------------------------------
# Generate a child script FILE with all initialization necessary to
# reuse the environment learned by the parent script, and make the
# file executable.  If COMMENT is supplied, it is inserted after the
# '#!' sequence but before initialization text begins.  After this
# macro, additional text can be appended to FILE to form the body of
# the child script.  The macro ends with non-zero status if the
# file could not be fully written (such as if the disk is full).
m4_ifdef([AS_INIT_GENERATED],
[m4_defun([_LT_GENERATED_FILE_INIT],[AS_INIT_GENERATED($@)])],
[m4_defun([_LT_GENERATED_FILE_INIT],
[m4_require([AS_PREPARE])]dnl
[m4_pushdef([AS_MESSAGE_LOG_FD])]dnl
[lt_write_fail=0
cat >$1 <<_ASEOF || lt_write_fail=1
```

```
#! $SHELL
# Generated by $as_me.
$2
SHELL=\${CONFIG_SHELL-$SHELL}
export SHELL
_ASEOF
cat >>$1 <<\_ASEOF || lt_write_fail=1
AS_SHELL_SANITIZE
_AS_PREPARE
exec AS_MESSAGE_FD>&1
_ASEOF
test 0 = "$lt_write_fail" && chmod +x $1[]dnl
m4_popdef([AS_MESSAGE_LOG_FD])])])# _LT_GENERATED_FILE_INIT

# LT_OUTPUT
# ---------
# This macro allows early generation of the libtool script (before
# AC_OUTPUT is called), incase it is used in configure for compilation
# tests.
AC_DEFUN([LT_OUTPUT],
[: ${CONFIG_LT=./config.lt}
AC_MSG_NOTICE([creating $CONFIG_LT])
_LT_GENERATED_FILE_INIT(["$CONFIG_LT"],
[# Run this file to recreate a libtool stub with the current configuration.])

cat >>"$CONFIG_LT" <<\_LTEOF
lt_cl_silent=false
exec AS_MESSAGE_LOG_FD>>config.log
{
 echo
 AS_BOX([Running $as_me.])
} >&AS_MESSAGE_LOG_FD

lt_cl_help="\
'$as_me' creates a local libtool stub from the current configuration,
for use in further configure time tests before the real libtool is
generated.

Usage: $[0] [[OPTIONS]]

 -h, --help      print this help, then exit
 -V, --version   print version number, then exit
 -q, --quiet     do not print progress messages
 -d, --debug     don't remove temporary files

Report bugs to <bug-libtool@gnu.org>."

lt_cl_version="\
m4_ifset([AC_PACKAGE_NAME], [AC_PACKAGE_NAME ])config.lt[]dnl
m4_ifset([AC_PACKAGE_VERSION], [ AC_PACKAGE_VERSION])
configured by $[0], generated by m4_PACKAGE_STRING.

Copyright (C) 2011 Free Software Foundation, Inc.
This config.lt script is free software; the Free Software Foundation
gives unlimited permision to copy, distribute and modify it."

while test 0 != $[#]
do
 case $[1] in
   --version | --v* | -V )
     echo "$lt_cl_version"; exit 0 ;;
   --help | --h* | -h )
     echo "$lt_cl_help"; exit 0 ;;
   --debug | --d* | -d )
     debug=: ;;
   --quiet | --q* | --silent | --s* | -q )
     lt_cl_silent=: ;;

   -*) AC_MSG_ERROR([unrecognized option: $[1]
Try '$[0] --help' for more information.]) ;;

   *) AC_MSG_ERROR([unrecognized argument: $[1]
Try '$[0] --help' for more information.]) ;;
 esac
 shift
done

if $lt_cl_silent; then
 exec AS_MESSAGE_FD>/dev/null
fi
_LTEOF

cat >>"$CONFIG_LT" <<_LTEOF
_LT_OUTPUT_LIBTOOL_COMMANDS_INIT
_LTEOF

cat >>"$CONFIG_LT" <<\_LTEOF
AC_MSG_NOTICE([creating $ofile])
_LT_OUTPUT_LIBTOOL_COMMANDS
AS_EXIT(0)
_LTEOF
chmod +x "$CONFIG_LT"

# configure is writing to config.log, but config.lt does its own redirection,
# appending to config.log, which fails on DOS, as config.log is still kept
# open by configure.  Here we exec the FD to /dev/null, effectively closing
# config.log, so it can be properly (re)opened and appended to by config.lt.
lt_cl_success=:
test yes = "$silent" &&
 lt_config_lt_args="$lt_config_lt_args --quiet"
exec AS_MESSAGE_LOG_FD>/dev/null
$SHELL "$CONFIG_LT" $lt_config_lt_args || lt_cl_success=false
exec AS_MESSAGE_LOG_FD>>config.log
$lt_cl_success || AS_EXIT(1)
])# LT_OUTPUT


# _LT_CONFIG(TAG)
# ---------------
# If TAG is the built-in tag, create an initial libtool script with a
# default configuration from the untagged config vars.  Otherwise add code
# to config.status for appending the configuration named by TAG from the
# matching tagged config vars.
m4_defun([_LT_CONFIG],
[m4_require([_LT_FILEUTILS_DEFAULTS])dnl
_LT_CONFIG_SAVE_COMMANDS([
 m4_define([_LT_TAG], m4_if([$1], [], [C], [$1]))dnl
 m4_if(_LT_TAG, [C], [
   # See if we are running on zsh, and set the options that allow our
   # commands through without removal of \ escapes.
   if test -n "${ZSH_VERSION+set}"; then
     setopt NO_GLOB_SUBST
   fi

   cfgfile=${ofile}T
   trap "$RM \"$cfgfile\"; exit 1" 1 2 15
   $RM "$cfgfile"

   cat <<_LT_EOF >> "$cfgfile"
#! $SHELL
# Generated automatically by $as_me ($PACKAGE) $VERSION
# NOTE: Changes made to this file will be lost: look at ltmain.sh.

# Provide generalized library-building support services.
# Written by Gordon Matzigkeit, 1996

_LT_COPYING
_LT_LIBTOOL_TAGS

# Configured defaults for sys_lib_dlsearch_path munging.
: \${LT_SYS_LIBRARY_PATH="$configure_time_lt_sys_library_path"}

# ### BEGIN LIBTOOL CONFIG
_LT_LIBTOOL_CONFIG_VARS
_LT_LIBTOOL_TAG_VARS
# ### END LIBTOOL CONFIG

_LT_EOF

   cat <<'_LT_EOF' >> "$cfgfile"

# ### BEGIN FUNCTIONS SHARED WITH CONFIGURE

_LT_PREPARE_MUNGE_PATH_LIST
_LT_PREPARE_CC_BASENAME

# ### END FUNCTIONS SHARED WITH CONFIGURE

_LT_EOF

 case $host_os in
 aix3*)
   cat <<\_LT_EOF >> "$cfgfile"
# AIX sometimes has problems with the GCC collect2 program.  For some
# reason, if we set the COLLECT_NAMES environment variable, the problems
# vanish in a puff of smoke.
if test set != "${COLLECT_NAMES+set}"; then
 COLLECT_NAMES=
 export COLLECT_NAMES
fi
_LT_EOF
   ;;
 esac

 _LT_PROG_LTMAIN

 # We use sed instead of cat because bash on DJGPP gets confused if
 # if finds mixed CR/LF and LF-only lines.  Since sed operates in
 # text mode, it properly converts lines to CR/LF.  This bash problem
 # is reportedly fixed, but why not run on old versions too?
 sed '$q' "$ltmain" >> "$cfgfile" \
    || (rm -f "$cfgfile"; exit 1)

  mv -f "$cfgfile" "$ofile" ||
   (rm -f "$ofile" && cp "$cfgfile" "$ofile" && rm -f "$cfgfile")
 chmod +x "$ofile"
],
[cat <<_LT_EOF >> "$ofile"

dnl Unfortunately we have to use $1 here, since _LT_TAG is not expanded
dnl in a comment (ie after a #).
# ### BEGIN LIBTOOL TAG CONFIG: $1
```

```
_LT_LIBTOOL_TAG_VARS(_LT_TAG)
# ### END LIBTOOL TAG CONFIG: $1
_LT_EOF
])dnl /m4_if
],
[m4_if([$1], [], [
   PACKAGE='$PACKAGE'
   VERSION='$VERSION'
   RM='$RM'
   ofile='$ofile'], [])
])dnl /_LT_CONFIG_SAVE_COMMANDS
])# _LT_CONFIG


# LT_SUPPORTED_TAG(TAG)
# ---------------------
# Trace this macro to discover what tags are supported by the libtool
# --tag option, using:
#    autoconf --trace 'LT_SUPPORTED_TAG:$1'
AC_DEFUN([LT_SUPPORTED_TAG], [])


# C support is built-in for now
m4_define([_LT_LANG_C_enabled], [])
m4_define([_LT_TAGS], [])


# LT_LANG(LANG)
# -------------
# Enable libtool support for the given language if not already enabled.
AC_DEFUN([LT_LANG],
[AC_BEFORE([$0], [LT_OUTPUT])dnl
m4_case([$1],
 [C],   [_LT_LANG(C)],
 [C++],  [_LT_LANG(CXX)],
 [Go],   [_LT_LANG(GO)],
 [Java],  [_LT_LANG(GCJ)],
 [Fortran 77],  [_LT_LANG(F77)],
 [Fortran],  [_LT_LANG(FC)],
 [Windows Resource], [_LT_LANG(RC)],
 [m4_ifdef([_LT_LANG_]$1[_CONFIG],
   [_LT_LANG($1)],
   [m4_fatal([$0: unsupported language: "$1"])])])dnl
])# LT_LANG


# _LT_LANG(LANGNAME)
# ------------------
```

```
m4_defun([_LT_LANG],
[m4_ifdef([_LT_LANG_]$1[_enabled], [],
 [LT_SUPPORTED_TAG([$1])dnl
 m4_append([_LT_TAGS], [$1 ])dnl
 m4_define([_LT_LANG_]$1[_enabled], [])dnl
 _LT_LANG_$1_CONFIG($1)])dnl
])# _LT_LANG


m4_ifndef([AC_PROG_GO], [
############################################################
# NOTE: This macro has been submitted for inclusion into   #
#  GNU Autoconf as AC_PROG_GO.  When it is available in    #
#  a released version of Autoconf we should remove this    #
#  macro and use it instead.                               #
############################################################
m4_defun([AC_PROG_GO],
[AC_LANG_PUSH(Go)dnl
AC_ARG_VAR([GOC],     [Go compiler command])dnl
AC_ARG_VAR([GOFLAGS], [Go compiler flags])dnl
_AC_ARG_VAR_LDFLAGS()dnl
AC_CHECK_TOOL(GOC, gccgo)
if test -z "$GOC"; then
 if test -n "$ac_tool_prefix"; then
   AC_CHECK_PROG(GOC, [${ac_tool_prefix}gccgo], [${ac_tool_prefix}gccgo])
 fi
fi
if test -z "$GOC"; then
 AC_CHECK_PROG(GOC, gccgo, gccgo, false)
fi
])#m4_defun
])#m4_ifndef


# _LT_LANG_DEFAULT_CONFIG
# -----------------------
m4_defun([_LT_LANG_DEFAULT_CONFIG],
[AC_PROVIDE_IFELSE([AC_PROG_CXX],
 [LT_LANG(CXX)],
 [m4_define([AC_PROG_CXX], defn([AC_PROG_CXX])[LT_LANG(CXX)])])

AC_PROVIDE_IFELSE([AC_PROG_F77],
 [LT_LANG(F77)],
 [m4_define([AC_PROG_F77], defn([AC_PROG_F77])[LT_LANG(F77)])])

AC_PROVIDE_IFELSE([AC_PROG_FC],
 [LT_LANG(FC)],
 [m4_define([AC_PROG_FC], defn([AC_PROG_FC])[LT_LANG(FC)])])
```

```
dnl The call to [A][M_PROG_GCJ] is quoted like that to stop aclocal
dnl pulling things in needlessly.
AC_PROVIDE_IFELSE([AC_PROG_GCJ],
 [LT_LANG(GCJ)],
 [AC_PROVIDE_IFELSE([A][M_PROG_GCJ],
   [LT_LANG(GCJ)],
   [AC_PROVIDE_IFELSE([LT_PROG_GCJ],
     [LT_LANG(GCJ)],
     [m4_ifdef([AC_PROG_GCJ],
 [m4_define([AC_PROG_GCJ], defn([AC_PROG_GCJ])[LT_LANG(GCJ)])])
      m4_ifdef([A][M_PROG_GCJ],
 [m4_define([A][M_PROG_GCJ], defn([A][M_PROG_GCJ])[LT_LANG(GCJ)])])
      m4_ifdef([LT_PROG_GCJ],
 [m4_define([LT_PROG_GCJ], defn([LT_PROG_GCJ])[LT_LANG(GCJ)])])])])])

AC_PROVIDE_IFELSE([AC_PROG_GO],
 [LT_LANG(GO)],
 [m4_define([AC_PROG_GO], defn([AC_PROG_GO])[LT_LANG(GO)])])

AC_PROVIDE_IFELSE([LT_PROG_RC],
 [LT_LANG(RC)],
 [m4_define([LT_PROG_RC], defn([LT_PROG_RC])[LT_LANG(RC)])])
])# _LT_LANG_DEFAULT_CONFIG

# Obsolete macros:
AU_DEFUN([AC_LIBTOOL_CXX], [LT_LANG(C++)])
AU_DEFUN([AC_LIBTOOL_F77], [LT_LANG(Fortran 77)])
AU_DEFUN([AC_LIBTOOL_FC], [LT_LANG(Fortran)])
AU_DEFUN([AC_LIBTOOL_GCJ], [LT_LANG(Java)])
AU_DEFUN([AC_LIBTOOL_RC], [LT_LANG(Windows Resource)])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AC_LIBTOOL_CXX], [])
dnl AC_DEFUN([AC_LIBTOOL_F77], [])
dnl AC_DEFUN([AC_LIBTOOL_FC], [])
dnl AC_DEFUN([AC_LIBTOOL_GCJ], [])
dnl AC_DEFUN([AC_LIBTOOL_RC], [])


# _LT_TAG_COMPILER
# ----------------
m4_defun([_LT_TAG_COMPILER],
[AC_REQUIRE([AC_PROG_CC])dnl

_LT_DECL([LTCC], [CC], [1], [A C compiler])dnl
_LT_DECL([LTCFLAGS], [CFLAGS], [1], [LTCC compiler flags])dnl
_LT_TAGDECL([CC], [compiler], [1], [A language specific compiler])dnl
_LT_TAGDECL([with_gcc], [GCC], [0], [Is the compiler the GNU compiler?])dnl
```

```
# If no C compiler was specified, use CC.
LTCC=${LTCC-"$CC"}

# If no C compiler flags were specified, use CFLAGS.
LTCFLAGS=${LTCFLAGS-"$CFLAGS"}

# Allow CC to be a program name with arguments.
compiler=$CC
])# _LT_TAG_COMPILER


# _LT_COMPILER_BOILERPLATE
# ------------------------
# Check for compiler boilerplate output or warnings with
# the simple compiler test code.
m4_defun([_LT_COMPILER_BOILERPLATE],
[m4_require([_LT_DECL_SED])dnl
ac_outfile=conftest.$ac_objext
echo "$lt_simple_compile_test_code" >conftest.$ac_ext
eval "$ac_compile" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d' >conftest.err
_lt_compiler_boilerplate=`cat conftest.err`
$RM conftest*
])# _LT_COMPILER_BOILERPLATE


# _LT_LINKER_BOILERPLATE
# ----------------------
# Check for linker boilerplate output or warnings with
# the simple link test code.
m4_defun([_LT_LINKER_BOILERPLATE],
[m4_require([_LT_DECL_SED])dnl
ac_outfile=conftest.$ac_objext
echo "$lt_simple_link_test_code" >conftest.$ac_ext
eval "$ac_link" 2>&1 >/dev/null | $SED '/^$/d; /^ *+/d' >conftest.err
_lt_linker_boilerplate=`cat conftest.err`
$RM -r conftest*
])# _LT_LINKER_BOILERPLATE

# _LT_REQUIRED_DARWIN_CHECKS
# -------------------------
m4_defun_once([_LT_REQUIRED_DARWIN_CHECKS],[
 case $host_os in
   rhapsody* | darwin*)
   AC_CHECK_TOOL([DSYMUTIL], [dsymutil], [:])
   AC_CHECK_TOOL([NMEDIT], [nmedit], [:])
   AC_CHECK_TOOL([LIPO], [lipo], [:])
   AC_CHECK_TOOL([OTOOL], [otool], [:])
```

```
  AC_CHECK_TOOL([OTOOL64], [otool64], [:])
  _LT_DECL([], [DSYMUTIL], [1],
    [Tool to manipulate archived DWARF debug symbol files on Mac OS X])
  _LT_DECL([], [NMEDIT], [1],
    [Tool to change global to local symbols on Mac OS X])
  _LT_DECL([], [LIPO], [1],
    [Tool to manipulate fat objects and archives on Mac OS X])
  _LT_DECL([], [OTOOL], [1],
    [ldd/readelf like tool for Mach-O binaries on Mac OS X])
  _LT_DECL([], [OTOOL64], [1],
    [ldd/readelf like tool for 64 bit Mach-O binaries on Mac OS X 10.4])

  AC_CACHE_CHECK([for -single_module linker flag],[lt_cv_apple_cc_single_mod],
    [lt_cv_apple_cc_single_mod=no
    if test -z "$LT_MULTI_MODULE"; then
 # By default we will add the -single_module flag. You can override
 # by either setting the environment variable LT_MULTI_MODULE
 # non-empty at configure time, or by adding -multi_module to the
 # link flags.
 rm -rf libconftest.dylib*
 echo "int foo(void){return 1;}" > conftest.c
 echo "$LTCC $LTCFLAGS $LDFLAGS -o libconftest.dylib \
-dynamiclib -Wl,-single_module conftest.c" >&AS_MESSAGE_LOG_FD
 $LTCC $LTCFLAGS $LDFLAGS -o libconftest.dylib \
   -dynamiclib -Wl,-single_module conftest.c 2>conftest.err
      _lt_result=$?
 # If there is a non-empty error log, and "single_module"
 # appears in it, assume the flag caused a linker warning
      if test -s conftest.err && $GREP single_module conftest.err; then
   cat conftest.err >&AS_MESSAGE_LOG_FD
 # Otherwise, if the output was created with a 0 exit code from
 # the compiler, it worked.
 elif test -f libconftest.dylib && test 0 = "$_lt_result"; then
   lt_cv_apple_cc_single_mod=yes
 else
   cat conftest.err >&AS_MESSAGE_LOG_FD
 fi
 rm -rf libconftest.dylib*
 rm -f conftest.*
    fi])

  AC_CACHE_CHECK([for -exported_symbols_list linker flag],
    [lt_cv_ld_exported_symbols_list],
    [lt_cv_ld_exported_symbols_list=no
    save_LDFLAGS=$LDFLAGS
    echo "_main" > conftest.sym
    LDFLAGS="$LDFLAGS -Wl,-exported_symbols_list,conftest.sym"
    AC_LINK_IFELSE([AC_LANG_PROGRAM([],[])],
 [lt_cv_ld_exported_symbols_list=yes],
 [lt_cv_ld_exported_symbols_list=no])
 LDFLAGS=$save_LDFLAGS
   ])

   AC_CACHE_CHECK([for -force_load linker flag],[lt_cv_ld_force_load],
     [lt_cv_ld_force_load=no
     cat > conftest.c << _LT_EOF
int forced_loaded() { return 2;}
_LT_EOF
     echo "$LTCC $LTCFLAGS -c -o conftest.o conftest.c" >&AS_MESSAGE_LOG_FD
     $LTCC $LTCFLAGS -c -o conftest.o conftest.c 2>&AS_MESSAGE_LOG_FD
     echo "$AR cru libconftest.a conftest.o" >&AS_MESSAGE_LOG_FD
     $AR cru libconftest.a conftest.o 2>&AS_MESSAGE_LOG_FD
     echo "$RANLIB libconftest.a" >&AS_MESSAGE_LOG_FD
     $RANLIB libconftest.a 2>&AS_MESSAGE_LOG_FD
     cat > conftest.c << _LT_EOF
int main() { return 0;}
_LT_EOF
     echo "$LTCC $LTCFLAGS $LDFLAGS -o conftest conftest.c -Wl,-force_load,./libconftest.a"
>&AS_MESSAGE_LOG_FD
     $LTCC $LTCFLAGS $LDFLAGS -o conftest conftest.c -Wl,-force_load,./libconftest.a 2>conftest.err
     _lt_result=$?
     if test -s conftest.err && $GREP force_load conftest.err; then
 cat conftest.err >&AS_MESSAGE_LOG_FD
     elif test -f conftest && test 0 = "$_lt_result" && $GREP forced_load conftest >/dev/null 2>&1; then
 lt_cv_ld_force_load=yes
     else
 cat conftest.err >&AS_MESSAGE_LOG_FD
     fi
       rm -f conftest.err libconftest.a conftest conftest.c
       rm -rf conftest.dSYM
   ])
   case $host_os in
   rhapsody* | darwin1.[[012]])
     _lt_dar_allow_undefined='$wl-undefined ${wl}suppress' ;;
   darwin1.*)
     _lt_dar_allow_undefined='$wl-flat_namespace $wl-undefined ${wl}suppress' ;;
   darwin*) # darwin 5.x on
     # if running on 10.5 or later, the deployment target defaults
     # to the OS version, if on x86, and 10.4, the deployment
     # target defaults to 10.4. Don't you love it?
     case ${MACOSX_DEPLOYMENT_TARGET-10.0},$host in
 10.0,*86*-darwin8*|10.0,*-darwin[[91]]*)
   _lt_dar_allow_undefined='$wl-undefined ${wl}dynamic_lookup' ;;
 10.[[012]][[,.]]*)
   _lt_dar_allow_undefined='$wl-flat_namespace $wl-undefined ${wl}suppress' ;;
 10.*)
```

```
    _lt_dar_allow_undefined='$wl-undefined ${wl}dynamic_lookup' ;;
      esac
    ;;
  esac
    if test yes = "$lt_cv_apple_cc_single_mod"; then
      _lt_dar_single_mod='$single_module'
    fi
    if test yes = "$lt_cv_ld_exported_symbols_list"; then
      _lt_dar_export_syms=' $wl-exported_symbols_list,$output_objdir/$libname-symbols.expsym'
    else
      _lt_dar_export_syms='~$NMEDIT -s $output_objdir/$libname-symbols.expsym $lib'
    fi
    if test : != "$DSYMUTIL" && test no = "$lt_cv_ld_force_load"; then
      _lt_dsymutil='~$DSYMUTIL $lib || :'
    else
      _lt_dsymutil=
    fi
    ;;
  esac
])


# _LT_DARWIN_LINKER_FEATURES([TAG])
# ---------------------------------
# Checks for linker and compiler features on darwin
m4_defun([_LT_DARWIN_LINKER_FEATURES],
[
  m4_require([_LT_REQUIRED_DARWIN_CHECKS])
  _LT_TAGVAR(archive_cmds_need_lc, $1)=no
  _LT_TAGVAR(hardcode_direct, $1)=no
  _LT_TAGVAR(hardcode_automatic, $1)=yes
  _LT_TAGVAR(hardcode_shlibpath_var, $1)=unsupported
  if test yes = "$lt_cv_ld_force_load"; then
    _LT_TAGVAR(whole_archive_flag_spec, $1)='`for conv in $convenience\"\"; do test  -n \"$conv\" &&
new_convenience=\"$new_convenience $wl-force_load,$conv\"; done; func_echo_all \"$new_convenience\"`'
    m4_case([$1], [F77], [_LT_TAGVAR(compiler_needs_object, $1)=yes],
                  [FC],  [_LT_TAGVAR(compiler_needs_object, $1)=yes])
  else
    _LT_TAGVAR(whole_archive_flag_spec, $1)=''
  fi
  _LT_TAGVAR(link_all_deplibs, $1)=yes
  _LT_TAGVAR(allow_undefined_flag, $1)=$_lt_dar_allow_undefined
  case $cc_basename in
     ifort*|nagfor*) _lt_dar_can_shared=yes ;;
     *) _lt_dar_can_shared=$GCC ;;
  esac
  if test yes = "$_lt_dar_can_shared"; then
    output_verbose_link_cmd=func_echo_all
```

```
   _LT_TAGVAR(archive_cmds, $1)="\$CC -dynamiclib \$allow_undefined_flag -o \$lib \$libobjs \$deplibs
\$compiler_flags -install_name \$rpath/\$soname \$verstring $_lt_dar_single_mod$_lt_dsymutil"
   _LT_TAGVAR(module_cmds, $1)="\$CC \$allow_undefined_flag -o \$lib -bundle \$libobjs \$deplibs
\$compiler_flags$_lt_dsymutil"
   _LT_TAGVAR(archive_expsym_cmds, $1)="sed 's|^|_|' < \$export_symbols > \$output_objdir/\$libname-
symbols.expsym~\$CC -dynamiclib \$allow_undefined_flag -o \$lib \$libobjs \$deplibs \$compiler_flags -
install_name \$rpath/\$soname \$verstring $_lt_dar_single_mod$_lt_dar_export_syms$_lt_dsymutil"
   _LT_TAGVAR(module_expsym_cmds, $1)="sed -e 's|^|_|' < \$export_symbols > \$output_objdir/\$libname-
symbols.expsym~\$CC \$allow_undefined_flag -o \$lib -bundle \$libobjs \$deplibs
\$compiler_flags$_lt_dar_export_syms$_lt_dsymutil"
   m4_if([$1], [CXX],
[   if test yes != "$lt_cv_apple_cc_single_mod"; then
     _LT_TAGVAR(archive_cmds, $1)="\$CC -r -keep_private_externs -nostdlib -o \$lib-master.o \$libobjs~\$CC -
dynamiclib \$allow_undefined_flag -o \$lib \$lib-master.o \$deplibs \$compiler_flags -install_name
\$rpath/\$soname \$verstring$_lt_dsymutil"
     _LT_TAGVAR(archive_expsym_cmds, $1)="sed 's|^|_|' < \$export_symbols > \$output_objdir/\$libname-
symbols.expsym~\$CC -r -keep_private_externs -nostdlib -o \$lib-master.o \$libobjs~\$CC -dynamiclib
\$allow_undefined_flag -o \$lib \$lib-master.o \$deplibs \$compiler_flags -install_name \$rpath/\$soname
\$verstring$_lt_dar_export_syms$_lt_dsymutil"
   fi
],[])
 else
 _LT_TAGVAR(ld_shlibs, $1)=no
 fi
])

# _LT_SYS_MODULE_PATH_AIX([TAGNAME])
# ----------------------------------
# Links a minimal program and checks the executable
# for the system default hardcoded library path. In most cases,
# this is /usr/lib:/lib, but when the MPI compilers are used
# the location of the communication and MPI libs are included too.
# If we don't find anything, use the default library path according
# to the aix ld manual.
# Store the results from the different compilers for each TAGNAME.
# Allow to override them for all tags through lt_cv_aix_libpath.
m4_defun([_LT_SYS_MODULE_PATH_AIX],
[m4_require([_LT_DECL_SED])dnl
if test set = "${lt_cv_aix_libpath+set}"; then
 aix_libpath=$lt_cv_aix_libpath
else
 AC_CACHE_VAL([_LT_TAGVAR([lt_cv_aix_libpath_], [$1])],
 [AC_LINK_IFELSE([AC_LANG_PROGRAM],[
 lt_aix_libpath_sed='[
     /Import File Strings/,/^$/ {
   /^0/ {
       s/^0  *\([^ ]*\) *$/\1/
       p
   }
     }]'
 _LT_TAGVAR([lt_cv_aix_libpath_], [$1])=`dump -H conftest$ac_exeext 2>/dev/null | $SED -n -e
"$lt_aix_libpath_sed"`
 # Check for a 64-bit object if we didn't find anything.
 if test -z "$_LT_TAGVAR([lt_cv_aix_libpath_], [$1])"; then
   _LT_TAGVAR([lt_cv_aix_libpath_], [$1])=`dump -HX64 conftest$ac_exeext 2>/dev/null | $SED -n -e
"$lt_aix_libpath_sed"`
 fi],[])
 if test -z "$_LT_TAGVAR([lt_cv_aix_libpath_], [$1])"; then
   _LT_TAGVAR([lt_cv_aix_libpath_], [$1])=/usr/lib:/lib
 fi
 ])
 aix_libpath=$_LT_TAGVAR([lt_cv_aix_libpath_], [$1])
fi
])# _LT_SYS_MODULE_PATH_AIX


# _LT_SHELL_INIT(ARG)
# --------------------
m4_define([_LT_SHELL_INIT],
[m4_divert_text([M4SH-INIT], [$1
])])# _LT_SHELL_INIT



# _LT_PROG_ECHO_BACKSLASH
# -----------------------
# Find how we can fake an echo command that does not interpret backslash.
# In particular, with Autoconf 2.60 or later we add some code to the start
# of the generated configure script that will find a shell with a builtin
# printf (that we can use as an echo command).
m4_defun([_LT_PROG_ECHO_BACKSLASH],
[ECHO='\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\'
ECHO=$ECHO$ECHO$ECHO$ECHO$ECHO
ECHO=$ECHO$ECHO$ECHO$ECHO$ECHO$ECHO

AC_MSG_CHECKING([how to print strings])
# Test print first, because it will be a builtin if present.
if test "X`( print -r -- -n ) 2>/dev/null`" = X-n && \
  test "X`print -r -- $ECHO 2>/dev/null`" = "X$ECHO"; then
 ECHO='print -r --'
elif test "X`printf %s $ECHO 2>/dev/null`" = "X$ECHO"; then
 ECHO='printf %s\n'
else
 # Use this function as a fallback that always works.
 func_fallback_echo ()
 {
   eval 'cat <<_LTECHO_EOF
$[]1
_LTECHO_EOF'
 }
 ECHO='func_fallback_echo'
fi

# func_echo_all arg...
# Invoke $ECHO with all args, space-separated.
func_echo_all ()
{
   $ECHO "$*"
}

case $ECHO in
 printf*) AC_MSG_RESULT([printf]) ;;
 print*) AC_MSG_RESULT([print -r]) ;;
 *) AC_MSG_RESULT([cat]) ;;
esac

m4_ifdef([_AS_DETECT_SUGGESTED],
[_AS_DETECT_SUGGESTED([
 test -n "${ZSH_VERSION+set}${BASH_VERSION+set}" || (
   ECHO='\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\'
   ECHO=$ECHO$ECHO$ECHO$ECHO$ECHO
   ECHO=$ECHO$ECHO$ECHO$ECHO$ECHO$ECHO
   PATH=/empty FPATH=/empty; export PATH FPATH
   test "X`printf %s $ECHO`" = "X$ECHO" \
     || test "X`print -r -- $ECHO`" = "X$ECHO" )])])

_LT_DECL([], [SHELL], [1], [Shell to use when invoking shell scripts])
_LT_DECL([], [ECHO], [1], [An echo program that protects backslashes])
])# _LT_PROG_ECHO_BACKSLASH


# _LT_WITH_SYSROOT
# ----------------
AC_DEFUN([_LT_WITH_SYSROOT],
[AC_MSG_CHECKING([for sysroot])
AC_ARG_WITH([sysroot],
[AS_HELP_STRING([--with-sysroot@<:@=DIR@:>@],
 [Search for dependent libraries within DIR (or the compiler's sysroot
  if not specified).])],
[], [with_sysroot=no])

dnl lt_sysroot will always be passed unquoted.  We quote it here
dnl in case the user passed a directory name.
lt_sysroot=
case $with_sysroot in #(
yes)
  if test yes = "$GCC"; then
    lt_sysroot=`$CC --print-sysroot 2>/dev/null`
  fi
  ;; #(
/*)
  lt_sysroot=`echo "$with_sysroot" | sed -e "$sed_quote_subst"`
  ;; #(
no|'')
  ;; #(
*)
  AC_MSG_RESULT([$with_sysroot])
  AC_MSG_ERROR([The sysroot must be an absolute path.])
  ;;
esac

AC_MSG_RESULT([${lt_sysroot:-no}])
_LT_DECL([], [lt_sysroot], [0], [The root where to search for ]dnl
[dependent libraries, and where our libraries should be installed.])])

# _LT_ENABLE_LOCK
# ---------------
m4_defun([_LT_ENABLE_LOCK],
[AC_ARG_ENABLE([libtool-lock],
 [AS_HELP_STRING([--disable-libtool-lock],
   [avoid locking (might break parallel builds)])])
test no = "$enable_libtool_lock" || enable_libtool_lock=yes

# Some flags need to be propagated to the compiler or linker for good
# libtool support.
case $host in
ia64-*-hpux*)
 # Find out what ABI is being produced by ac_compile, and set mode
 # options accordingly.
 echo 'int i;' > conftest.$ac_ext
 if AC_TRY_EVAL(ac_compile); then
   case `/usr/bin/file conftest.$ac_objext` in
     *ELF-32*)
 HPUX_IA64_MODE=32
 ;;
     *ELF-64*)
 HPUX_IA64_MODE=64
 ;;
   esac
 fi
 rm -rf conftest*
 ;;
```

```
*-*-irix6*)
 # Find out what ABI is being produced by ac_compile, and set linker
 # options accordingly.
 echo '[#]line '$LINENO' "configure"' > conftest.$ac_ext
 if AC_TRY_EVAL(ac_compile); then
   if test yes = "$lt_cv_prog_gnu_ld"; then
     case `/usr/bin/file conftest.$ac_objext` in
 *32-bit*)
   LD="${LD-ld} -melf32bsmip"
   ;;
 *N32*)
   LD="${LD-ld} -melf32bmipn32"
   ;;
 *64-bit*)
   LD="${LD-ld} -melf64bmip"
 ;;
     esac
   else
     case `/usr/bin/file conftest.$ac_objext` in
 *32-bit*)
   LD="${LD-ld} -32"
   ;;
 *N32*)
   LD="${LD-ld} -n32"
   ;;
 *64-bit*)
   LD="${LD-ld} -64"
   ;;
     esac
   fi
 fi
 rm -rf conftest*
 ;;

mips64*-*linux*)
 # Find out what ABI is being produced by ac_compile, and set linker
 # options accordingly.
 echo '[#]line '$LINENO' "configure"' > conftest.$ac_ext
 if AC_TRY_EVAL(ac_compile); then
   emul=elf
   case `/usr/bin/file conftest.$ac_objext` in
     *32-bit*)
 emul="${emul}32"
 ;;
     *64-bit*)
 emul="${emul}64"
 ;;
   esac
```

```
  case `/usr/bin/file conftest.$ac_objext` in
    *MSB*)
emul="${emul}btsmip"
;;
    *LSB*)
emul="${emul}ltsmip"
;;
  esac
  case `/usr/bin/file conftest.$ac_objext` in
    *N32*)
emul="${emul}n32"
;;
  esac
  LD="${LD-ld} -m $emul"
fi
rm -rf conftest*
;;

x86_64-*kfreebsd*-gnu|x86_64-*linux*|powerpc*-*linux*| \
s390*-*linux*|s390*-*tpf*|sparc*-*linux*)
# Find out what ABI is being produced by ac_compile, and set linker
# options accordingly.  Note that the listed cases only cover the
# situations where additional linker options are needed (such as when
# doing 32-bit compilation for a host where ld defaults to 64-bit, or
# vice versa); the common cases where no linker options are needed do
# not appear in the list.
echo 'int i;' > conftest.$ac_ext
if AC_TRY_EVAL(ac_compile); then
  case `/usr/bin/file conftest.o` in
    *32-bit*)
case $host in
  x86_64-*kfreebsd*-gnu)
    LD="${LD-ld} -m elf_i386_fbsd"
    ;;
  x86_64-*linux*)
    case `/usr/bin/file conftest.o` in
      *x86-64*)
 LD="${LD-ld} -m elf32_x86_64"
 ;;
      *)
 LD="${LD-ld} -m elf_i386"
 ;;
    esac
    ;;
  powerpc64le-*linux*)
    LD="${LD-ld} -m elf32lppclinux"
    ;;
  powerpc64-*linux*)
```

```
   LD="${LD-ld} -m elf32ppclinux"
   ;;
  s390x-*linux*)
   LD="${LD-ld} -m elf_s390"
   ;;
  sparc64-*linux*)
   LD="${LD-ld} -m elf32_sparc"
   ;;
 esac
 ;;
   *64-bit*)
 case $host in
  x86_64-*kfreebsd*-gnu)
   LD="${LD-ld} -m elf_x86_64_fbsd"
   ;;
  x86_64-*linux*)
   LD="${LD-ld} -m elf_x86_64"
   ;;
  powerpcle-*linux*)
   LD="${LD-ld} -m elf64lppc"
   ;;
  powerpc-*linux*)
   LD="${LD-ld} -m elf64ppc"
   ;;
  s390*-*linux*|s390*-*tpf*)
   LD="${LD-ld} -m elf64_s390"
   ;;
  sparc*-*linux*)
   LD="${LD-ld} -m elf64_sparc"
   ;;
 esac
 ;;
  esac
 fi
 rm -rf conftest*
 ;;

*-*-sco3.2v5*)
 # On SCO OpenServer 5, we need -belf to get full-featured binaries.
 SAVE_CFLAGS=$CFLAGS
 CFLAGS="$CFLAGS -belf"
 AC_CACHE_CHECK([whether the C compiler needs -belf], lt_cv_cc_needs_belf,
  [AC_LANG_PUSH(C)
   AC_LINK_IFELSE([AC_LANG_PROGRAM([[]],[[]])],[lt_cv_cc_needs_belf=yes],[lt_cv_cc_needs_belf=no])
   AC_LANG_POP])
 if test yes != "$lt_cv_cc_needs_belf"; then
  # this is probably gcc 2.8.0, egcs 1.0 or newer; no need for -belf
  CFLAGS=$SAVE_CFLAGS
 fi
 ;;
*-*solaris*)
 # Find out what ABI is being produced by ac_compile, and set linker
 # options accordingly.
 echo 'int i;' > conftest.$ac_ext
 if AC_TRY_EVAL(ac_compile); then
   case `/usr/bin/file conftest.o` in
   *64-bit*)
     case $lt_cv_prog_gnu_ld in
     yes*)
       case $host in
       i?86-*-solaris*|x86_64-*-solaris*)
         LD="${LD-ld} -m elf_x86_64"
         ;;
       sparc*-*-solaris*)
         LD="${LD-ld} -m elf64_sparc"
         ;;
       esac
       # GNU ld 2.21 introduced _sol2 emulations.  Use them if available.
       if ${LD-ld} -V | grep _sol2 >/dev/null 2>&1; then
         LD=${LD-ld}_sol2
       fi
       ;;
     *)
 if ${LD-ld} -64 -r -o conftest2.o conftest.o >/dev/null 2>&1; then
   LD="${LD-ld} -64"
 fi
 ;;
     esac
     ;;
   esac
 fi
 rm -rf conftest*
 ;;
esac

need_locks=$enable_libtool_lock
])# _LT_ENABLE_LOCK


# _LT_PROG_AR
# -----------
m4_defun([_LT_PROG_AR],
[AC_CHECK_TOOLS(AR, [ar], false)
: ${AR=ar}
: ${AR_FLAGS=cru}
_LT_DECL([], [AR], [1], [The archiver])
_LT_DECL([], [AR_FLAGS], [1], [Flags to create an archive])

AC_CACHE_CHECK([for archiver @FILE support], [lt_cv_ar_at_file],
 [lt_cv_ar_at_file=no
  AC_COMPILE_IFELSE([AC_LANG_PROGRAM],
    [echo conftest.$ac_objext > conftest.lst
     lt_ar_try='$AR $AR_FLAGS libconftest.a @conftest.lst >&AS_MESSAGE_LOG_FD'
     AC_TRY_EVAL([lt_ar_try])
     if test 0 -eq "$ac_status"; then
 # Ensure the archiver fails upon bogus file names.
 rm -f conftest.$ac_objext libconftest.a
 AC_TRY_EVAL([lt_ar_try])
 if test 0 -ne "$ac_status"; then
         lt_cv_ar_at_file=@
       fi
     fi
     rm -f conftest.* libconftest.a
    ])
 ])

if test no = "$lt_cv_ar_at_file"; then
 archiver_list_spec=
else
 archiver_list_spec=$lt_cv_ar_at_file
fi
_LT_DECL([], [archiver_list_spec], [1],
 [How to feed a file listing to the archiver])
])# _LT_PROG_AR


# _LT_CMD_OLD_ARCHIVE
# -------------------
m4_defun([_LT_CMD_OLD_ARCHIVE],
[_LT_PROG_AR

AC_CHECK_TOOL(STRIP, strip, :)
test -z "$STRIP" && STRIP=:
_LT_DECL([], [STRIP], [1], [A symbol stripping program])

AC_CHECK_TOOL(RANLIB, ranlib, :)
test -z "$RANLIB" && RANLIB=:
_LT_DECL([], [RANLIB], [1],
   [Commands used to install an old-style archive])

# Determine commands to create old-style static archives.
old_archive_cmds='$AR $AR_FLAGS $oldlib$oldobjs'
old_postinstall_cmds='chmod 644 $oldlib'
old_postuninstall_cmds=
if test -n "$RANLIB"; then
 case $host_os in
 bitrig* | openbsd*)
   old_postinstall_cmds="$old_postinstall_cmds~\$RANLIB -t \$tool_oldlib"
   ;;
 *)
   old_postinstall_cmds="$old_postinstall_cmds~\$RANLIB \$tool_oldlib"
   ;;
 esac
 old_archive_cmds="$old_archive_cmds~\$RANLIB \$tool_oldlib"
fi

case $host_os in
 darwin*)
   lock_old_archive_extraction=yes ;;
 *)
   lock_old_archive_extraction=no ;;
esac
_LT_DECL([], [old_postinstall_cmds], [2])
_LT_DECL([], [old_postuninstall_cmds], [2])
_LT_TAGDECL([], [old_archive_cmds], [2],
   [Commands used to build an old-style archive])
_LT_DECL([], [lock_old_archive_extraction], [0],
   [Whether to use a lock for old archive extraction])
])# _LT_CMD_OLD_ARCHIVE


# _LT_COMPILER_OPTION(MESSAGE, VARIABLE-NAME, FLAGS,
#  [OUTPUT-FILE], [ACTION-SUCCESS], [ACTION-FAILURE])
# ----------------------------------------------------------------
# Check whether the given compiler option works
AC_DEFUN([_LT_COMPILER_OPTION],
[m4_require([_LT_FILEUTILS_DEFAULTS])dnl
m4_require([_LT_DECL_SED])dnl
AC_CACHE_CHECK([$1], [$2],
 [$2=no
  m4_if([$4], , [ac_outfile=conftest.$ac_objext], [ac_outfile=$4])
  echo "$lt_simple_compile_test_code" > conftest.$ac_ext
  lt_compiler_flag="$3"  ## exclude from sc_useless_quotes_in_assignment
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "conftest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and begins
  # with a dollar sign (not a hyphen), so the echo should work correctly.
  # The option is referenced via a variable to avoid confusing sed.
  lt_compile=`echo "$ac_compile" | $SED \
  -e 's:.*FLAGS}\{0,1\} :&$lt_compiler_flag :; t' \
  -e 's: [[^ ]]*conftest\.: $lt_compiler_flag&:; t' \
  -e 's:$: $lt_compiler_flag:'`
  (eval echo "\"\$as_me:$LINENO: $lt_compile\"" >&AS_MESSAGE_LOG_FD)
  (eval "$lt_compile" 2>conftest.err)
  ac_status=$?
  cat conftest.err >&AS_MESSAGE_LOG_FD
  echo "$as_me:$LINENO: \$? = $ac_status" >&AS_MESSAGE_LOG_FD
  if (exit $ac_status) && test -s "$ac_outfile"; then
    # The compiler can only warn and ignore the option if not recognized
    # So say no if there are warnings other than the usual output.
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' >conftest.exp
    $SED '/^$/d; /^ *+/d' conftest.err >conftest.er2
    if test ! -s conftest.er2 || diff conftest.exp conftest.er2 >/dev/null; then
      $2=yes
    fi
  fi
  $RM conftest*
])

if test yes = "[$]$2"; then
   m4_if([$5], , :, [$5])
else
   m4_if([$6], , :, [$6])
fi
])# _LT_COMPILER_OPTION

# Old name:
AU_ALIAS([AC_LIBTOOL_COMPILER_OPTION], [_LT_COMPILER_OPTION])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AC_LIBTOOL_COMPILER_OPTION], [])


# _LT_LINKER_OPTION(MESSAGE, VARIABLE-NAME, FLAGS,
#                  [ACTION-SUCCESS], [ACTION-FAILURE])
# ----------------------------------------------------
# Check whether the given linker option works
AC_DEFUN([_LT_LINKER_OPTION],
[m4_require([_LT_FILEUTILS_DEFAULTS])dnl
m4_require([_LT_DECL_SED])dnl
AC_CACHE_CHECK([$1], [$2],
 [$2=no
  save_LDFLAGS=$LDFLAGS
  LDFLAGS="$LDFLAGS $3"
  echo "$lt_simple_link_test_code" > conftest.$ac_ext
  if (eval $ac_link 2>conftest.err) && test -s conftest$ac_exeext; then
    # The linker can only warn and ignore the option if not recognized
    # So say no if there are warnings
    if test -s conftest.err; then
      # Append any errors to the config.log.
      cat conftest.err 1>&AS_MESSAGE_LOG_FD
      $ECHO "$_lt_linker_boilerplate" | $SED '/^$/d' > conftest.exp
      $SED '/^$/d; /^ *+/d' conftest.err >conftest.er2
      if diff conftest.exp conftest.er2 >/dev/null; then
        $2=yes
      fi
    else
      $2=yes
    fi
  fi
  $RM -r conftest*
  LDFLAGS=$save_LDFLAGS
])

if test yes = "[$]$2"; then
   m4_if([$4], , :, [$4])
else
   m4_if([$5], , :, [$5])
fi
])# _LT_LINKER_OPTION

# Old name:
AU_ALIAS([AC_LIBTOOL_LINKER_OPTION], [_LT_LINKER_OPTION])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AC_LIBTOOL_LINKER_OPTION], [])


# LT_CMD_MAX_LEN
#---------------
AC_DEFUN([LT_CMD_MAX_LEN],
[AC_REQUIRE([AC_CANONICAL_HOST])dnl
# find the maximum length of command line arguments
AC_MSG_CHECKING([the maximum length of command line arguments])
AC_CACHE_VAL([lt_cv_sys_max_cmd_len], [dnl
 i=0
 teststring=ABCD

 case $build_os in
 msdosdjgpp*)
   # On DJGPP, this test can blow up pretty badly due to problems in libc
   # (any single argument exceeding 2000 bytes causes a buffer overrun
   # during glob expansion).  Even if it were fixed, the result of this
   # check would be larger than it should be.
   lt_cv_sys_max_cmd_len=12288;    # 12K is about right
   ;;

 gnu*)
   # Under GNU Hurd, this test is not required because there is
   # no limit to the length of command line arguments.
   # Libtool will interpret -1 as no limit whatsoever
   lt_cv_sys_max_cmd_len=-1;
   ;;

 cygwin* | mingw* | cegcc*)
   # On Win9x/ME, this test blows up -- it succeeds, but takes
   # about 5 minutes as the teststring grows exponentially.
   # Worse, since 9x/ME are not pre-emptively multitasking,
   # you end up with a "frozen" computer, even though with patience
   # the test eventually succeeds (with a max line length of 256k).
   # Instead, let's just punt: use the minimum linelength reported by
   # all of the supported platforms: 8192 (on NT/2K/XP).
   lt_cv_sys_max_cmd_len=8192;
   ;;

 mint*)
   # On MiNT this can take a long time and run out of memory.
   lt_cv_sys_max_cmd_len=8192;
   ;;

 amigaos*)
   # On AmigaOS with pdksh, this test takes hours, literally.
   # So we just punt and use a minimum line length of 8192.
   lt_cv_sys_max_cmd_len=8192;
   ;;

 bitrig* | darwin* | dragonfly* | freebsd* | netbsd* | openbsd*)
   # This has been around since 386BSD, at least.  Likely further.
   if test -x /sbin/sysctl; then
     lt_cv_sys_max_cmd_len=`/sbin/sysctl -n kern.argmax`
   elif test -x /usr/sbin/sysctl; then
     lt_cv_sys_max_cmd_len=`/usr/sbin/sysctl -n kern.argmax`
   else
     lt_cv_sys_max_cmd_len=65536 # usable default for all BSDs
   fi
   # And add a safety zone
   lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 4`
   lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \* 3`
   ;;

 interix*)
   # We know the value 262144 and hardcode it with a safety zone (like BSD)
   lt_cv_sys_max_cmd_len=196608
   ;;

 os2*)
   # The test takes a long time on OS/2.
   lt_cv_sys_max_cmd_len=8192
   ;;

 osf*)
   # Dr. Hans Ekkehard Plesser reports seeing a kernel panic running configure
   # due to this test when exec_disable_arg_limit is 1 on Tru64. It is not
   # nice to cause kernel panics so lets avoid the loop below.
   # First set a reasonable default.
   lt_cv_sys_max_cmd_len=16384
   #
   if test -x /sbin/sysconfig; then
     case `/sbin/sysconfig -q proc exec_disable_arg_limit` in
       *1*) lt_cv_sys_max_cmd_len=-1 ;;
     esac
   fi
   ;;
 sco3.2v5*)
   lt_cv_sys_max_cmd_len=102400
   ;;
 sysv5* | sco5v6* | sysv4.2uw2*)
   kargmax=`grep ARG_MAX /etc/conf/cf.d/stune 2>/dev/null`
   if test -n "$kargmax"; then
     lt_cv_sys_max_cmd_len=`echo $kargmax | sed 's/.*[[  ]]//'`
   else
     lt_cv_sys_max_cmd_len=32768
   fi
   ;;
 *)
   lt_cv_sys_max_cmd_len=`(getconf ARG_MAX) 2> /dev/null`
   if test -n "$lt_cv_sys_max_cmd_len" && \
      test undefined != "$lt_cv_sys_max_cmd_len"; then
     lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 4`
     lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \* 3`
   else
     # Make teststring a little bigger before we do anything with it.
     # a 1K string should be a reasonable start.
     for i in 1 2 3 4 5 6 7 8; do
       teststring=$teststring$teststring
     done
     SHELL=${SHELL-${CONFIG_SHELL-/bin/sh}}
     # If test is not a shell built-in, we'll probably end up computing a
     # maximum length that is only half of the actual maximum length, but
     # we can't tell.
     while { test X`env echo "$teststring$teststring" 2>/dev/null` \
         = "X$teststring$teststring"; } >/dev/null 2>&1 &&
       test 17 != "$i" # 1/2 MB should be enough
     do
       i=`expr $i + 1`
       teststring=$teststring$teststring
     done
     # Only check the string length outside the loop.
     lt_cv_sys_max_cmd_len=`expr "X$teststring" : ".*" 2>&1`
     teststring=
     # Add a significant safety factor because C++ compilers can tack on
     # massive amounts of additional arguments before passing them to the
     # linker.  It appears as though 1/2 is a usable value.
     lt_cv_sys_max_cmd_len=`expr $lt_cv_sys_max_cmd_len \/ 2`
   fi
   ;;
 esac
])
if test -n "$lt_cv_sys_max_cmd_len"; then
 AC_MSG_RESULT($lt_cv_sys_max_cmd_len)
else
 AC_MSG_RESULT(none)
fi
max_cmd_len=$lt_cv_sys_max_cmd_len
_LT_DECL([], [max_cmd_len], [0],
   [What is the maximum length of a command?])
])# LT_CMD_MAX_LEN

# Old name:
AU_ALIAS([AC_LIBTOOL_SYS_MAX_CMD_LEN], [LT_CMD_MAX_LEN])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AC_LIBTOOL_SYS_MAX_CMD_LEN], [])


# _LT_HEADER_DLFCN
# ----------------
m4_defun([_LT_HEADER_DLFCN],
[AC_CHECK_HEADERS([dlfcn.h], [], [], [AC_INCLUDES_DEFAULT])dnl
])# _LT_HEADER_DLFCN


# _LT_TRY_DLOPEN_SELF (ACTION-IF-TRUE, ACTION-IF-TRUE-W-USCORE,
#                      ACTION-IF-FALSE, ACTION-IF-CROSS-COMPILING)
# ----------------------------------------------------------------
m4_defun([_LT_TRY_DLOPEN_SELF],
[m4_require([_LT_HEADER_DLFCN])dnl
if test yes = "$cross_compiling"; then :
 [$4]
else
 lt_dlunknown=0; lt_dlno_uscore=1; lt_dlneed_uscore=2
 lt_status=$lt_dlunknown
 cat > conftest.$ac_ext <<_LT_EOF
[#line $LINENO "configure"
```

```
#include "confdefs.h"

#if HAVE_DLFCN_H
#include <dlfcn.h>
#endif

#include <stdio.h>

#ifdef RTLD_GLOBAL
#  define LT_DLGLOBAL  RTLD_GLOBAL
#else
#  ifdef DL_GLOBAL
#    define LT_DLGLOBAL  DL_GLOBAL
#  else
#    define LT_DLGLOBAL  0
#  endif
#endif

/* We may have to define LT_DLLAZY_OR_NOW in the command line if we
   find out it does not work in some platform. */
#ifndef LT_DLLAZY_OR_NOW
#  ifdef RTLD_LAZY
#    define LT_DLLAZY_OR_NOW  RTLD_LAZY
#  else
#    ifdef DL_LAZY
#      define LT_DLLAZY_OR_NOW  DL_LAZY
#    else
#      ifdef RTLD_NOW
#        define LT_DLLAZY_OR_NOW RTLD_NOW
#      else
#        ifdef DL_NOW
#          define LT_DLLAZY_OR_NOW DL_NOW
#        else
#          define LT_DLLAZY_OR_NOW 0
#        endif
#      endif
#    endif
#  endif
#endif

/* When -fvisibility=hidden is used, assume the code has been annotated
   correspondingly for the symbols needed.  */
#if defined __GNUC__ && (((__GNUC__ == 3) && (__GNUC_MINOR__ >= 3)) || (__GNUC__ > 3))
int fnord () __attribute__((visibility("default")));
#endif

int fnord () { return 42; }
int main ()
```

```
{
 void *self = dlopen (0, LT_DLGLOBAL|LT_DLLAZY_OR_NOW);
 int status = $lt_dlunknown;

 if (self)
   {
     if (dlsym (self,"fnord"))       status = $lt_dlno_uscore;
     else
       {
   if (dlsym( self,"_fnord"))  status = $lt_dlneed_uscore;
         else puts (dlerror ());
 }
     /* dlclose (self); */
   }
 else
   puts (dlerror ());

 return status;
}]
_LT_EOF
 if AC_TRY_EVAL(ac_link) && test -s "conftest$ac_exeext" 2>/dev/null; then
   (./conftest; exit; ) >&AS_MESSAGE_LOG_FD 2>/dev/null
   lt_status=$?
   case x$lt_status in
     x$lt_dlno_uscore) $1 ;;
     x$lt_dlneed_uscore) $2 ;;
     x$lt_dlunknown|x*) $3 ;;
   esac
 else :
   # compilation failed
   $3
 fi
fi
rm -fr conftest*
])# _LT_TRY_DLOPEN_SELF


# LT_SYS_DLOPEN_SELF
# ------------------
AC_DEFUN([LT_SYS_DLOPEN_SELF],
[m4_require([_LT_HEADER_DLFCN])dnl
if test yes != "$enable_dlopen"; then
 enable_dlopen=unknown
 enable_dlopen_self=unknown
 enable_dlopen_self_static=unknown
else
 lt_cv_dlopen=no
 lt_cv_dlopen_libs=
 case $host_os in
 beos*)
   lt_cv_dlopen=load_add_on
   lt_cv_dlopen_libs=
   lt_cv_dlopen_self=yes
   ;;

 mingw* | pw32* | cegcc*)
   lt_cv_dlopen=LoadLibrary
   lt_cv_dlopen_libs=
   ;;

 cygwin*)
   lt_cv_dlopen=dlopen
   lt_cv_dlopen_libs=
   ;;

 darwin*)
   # if libdl is installed we need to link against it
   AC_CHECK_LIB([dl], [dlopen],
  [lt_cv_dlopen=dlopen lt_cv_dlopen_libs=-ldl],[
   lt_cv_dlopen=dyld
   lt_cv_dlopen_libs=
   lt_cv_dlopen_self=yes
   ])
   ;;

 tpf*)
   # Don't try to run any link tests for TPF.  We know it's impossible
   # because TPF is a cross-compiler, and we know how we open DSOs.
   lt_cv_dlopen=dlopen
   lt_cv_dlopen_libs=
   lt_cv_dlopen_self=no
   ;;

 *)
   AC_CHECK_FUNC([shl_load],
   [lt_cv_dlopen=shl_load],
     [AC_CHECK_LIB([dld], [shl_load],
     [lt_cv_dlopen=shl_load lt_cv_dlopen_libs=-ldld],
 [AC_CHECK_FUNC([dlopen],
       [lt_cv_dlopen=dlopen],
   [AC_CHECK_LIB([dl], [dlopen],
  [lt_cv_dlopen=dlopen lt_cv_dlopen_libs=-ldl],
     [AC_CHECK_LIB([svld], [dlopen],
    [lt_cv_dlopen=dlopen lt_cv_dlopen_libs=-lsvld],
       [AC_CHECK_LIB([dld], [dld_link],
      [lt_cv_dlopen=dld_link lt_cv_dlopen_libs=-ldld])
       ])
     ])
   ])
 ])
     ])
   ;;
 esac

 if test no = "$lt_cv_dlopen"; then
   enable_dlopen=no
 else
   enable_dlopen=yes
 fi

 case $lt_cv_dlopen in
 dlopen)
   save_CPPFLAGS=$CPPFLAGS
   test yes = "$ac_cv_header_dlfcn_h" && CPPFLAGS="$CPPFLAGS -DHAVE_DLFCN_H"

   save_LDFLAGS=$LDFLAGS
   wl=$lt_prog_compiler_wl eval LDFLAGS=\"\$LDFLAGS $export_dynamic_flag_spec\"

   save_LIBS=$LIBS
   LIBS="$lt_cv_dlopen_libs $LIBS"

   AC_CACHE_CHECK([whether a program can dlopen itself],
   lt_cv_dlopen_self, [dnl
   _LT_TRY_DLOPEN_SELF(
     lt_cv_dlopen_self=yes, lt_cv_dlopen_self=yes,
     lt_cv_dlopen_self=no, lt_cv_dlopen_self=cross)
   ])

   if test yes = "$lt_cv_dlopen_self"; then
     wl=$lt_prog_compiler_wl eval LDFLAGS=\"\$LDFLAGS $lt_prog_compiler_static\"
     AC_CACHE_CHECK([whether a statically linked program can dlopen itself],
   lt_cv_dlopen_self_static, [dnl
   _LT_TRY_DLOPEN_SELF(
     lt_cv_dlopen_self_static=yes, lt_cv_dlopen_self_static=yes,
     lt_cv_dlopen_self_static=no,  lt_cv_dlopen_self_static=cross)
     ])
   fi

   CPPFLAGS=$save_CPPFLAGS
   LDFLAGS=$save_LDFLAGS
   LIBS=$save_LIBS
   ;;
 esac
 case $lt_cv_dlopen_self in
 yes|no) enable_dlopen_self=$lt_cv_dlopen_self ;;
 *) enable_dlopen_self=unknown ;;
 esac

 case $lt_cv_dlopen_self_static in
 yes|no) enable_dlopen_self_static=$lt_cv_dlopen_self_static ;;
 *) enable_dlopen_self_static=unknown ;;
 esac
fi
_LT_DECL([dlopen_support], [enable_dlopen], [0],
  [Whether dlopen is supported])
_LT_DECL([dlopen_self], [enable_dlopen_self], [0],
  [Whether dlopen of programs is supported])
_LT_DECL([dlopen_self_static], [enable_dlopen_self_static], [0],
  [Whether dlopen of statically linked programs is supported])
])# LT_SYS_DLOPEN_SELF

# Old name:
AU_ALIAS([AC_LIBTOOL_DLOPEN_SELF], [LT_SYS_DLOPEN_SELF])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AC_LIBTOOL_DLOPEN_SELF], [])


# _LT_COMPILER_C_O([TAGNAME])
# ---------------------------
# Check to see if options -c and -o are simultaneously supported by compiler.
# This macro does not hard code the compiler like AC_PROG_CC_C_O.
m4_defun([_LT_COMPILER_C_O],
[m4_require([_LT_DECL_SED])dnl
m4_require([_LT_FILEUTILS_DEFAULTS])dnl
m4_require([_LT_TAG_COMPILER])dnl
AC_CACHE_CHECK([if $compiler supports -c -o file.$ac_objext],
 [_LT_TAGVAR(lt_cv_prog_compiler_c_o, $1)],
 [_LT_TAGVAR(lt_cv_prog_compiler_c_o, $1)=no
  $RM -r conftest 2>/dev/null
  mkdir conftest
  cd conftest
  mkdir out
  echo "$lt_simple_compile_test_code" > conftest.$ac_ext

  lt_compiler_flag="-o out/conftest2.$ac_objext"
  # Insert the option either (1) after the last *FLAGS variable, or
  # (2) before a word containing "conftest.", or (3) at the end.
  # Note that $ac_compile itself does not contain backslashes and begins
  # with a dollar sign (not a hyphen), so the echo should work correctly.
  lt_compile=`echo "$ac_compile" | $SED \
  -e 's:.*FLAGS}\{0,1\} :&$lt_compiler_flag :; t' \
  -e 's: [[^ ]]*conftest\.: $lt_compiler_flag&:; t' \
  -e 's:$: $lt_compiler_flag:'`
  (eval echo "\"\$as_me:$LINENO: $lt_compile\"" >&AS_MESSAGE_LOG_FD)
  (eval "$lt_compile" 2>out/conftest.err)
  ac_status=$?
  cat out/conftest.err >&AS_MESSAGE_LOG_FD
  echo "$as_me:$LINENO: \$? = $ac_status" >&AS_MESSAGE_LOG_FD
  if (exit $ac_status) && test -s out/conftest2.$ac_objext
  then
    # The compiler can only warn and ignore the option if not recognized
    # So say no if there are warnings
    $ECHO "$_lt_compiler_boilerplate" | $SED '/^$/d' > out/conftest.exp
    $SED '/^$/d; /^ *+/d' out/conftest.err >out/conftest.er2
    if test ! -s out/conftest.er2 || diff out/conftest.exp out/conftest.er2 >/dev/null; then
      _LT_TAGVAR(lt_cv_prog_compiler_c_o, $1)=yes
    fi
  fi
  chmod u+w . 2>&AS_MESSAGE_LOG_FD
  $RM conftest*
  # SGI C++ compiler will create directory out/ii_files/ for
  # template instantiation
  test -d out/ii_files && $RM out/ii_files/* && rmdir out/ii_files
  $RM out/* && rmdir out
  cd ..
  $RM -r conftest
  $RM conftest*
])
_LT_TAGDECL([compiler_c_o], [lt_cv_prog_compiler_c_o], [1],
 [Does compiler simultaneously support -c and -o options?])
])# _LT_COMPILER_C_O


# _LT_COMPILER_FILE_LOCKS([TAGNAME])
# ----------------------------------
# Check to see if we can do hard links to lock some files if needed
m4_defun([_LT_COMPILER_FILE_LOCKS],
[m4_require([_LT_ENABLE_LOCK])dnl
m4_require([_LT_FILEUTILS_DEFAULTS])dnl
_LT_COMPILER_C_O([$1])

hard_links=nottested
if test no = "$_LT_TAGVAR(lt_cv_prog_compiler_c_o, $1)" && test no != "$need_locks"; then
 # do not overwrite the value of need_locks provided by the user
 AC_MSG_CHECKING([if we can lock with hard links])
 hard_links=yes
 $RM conftest*
 ln conftest.a conftest.b 2>/dev/null && hard_links=no
 touch conftest.a
 ln conftest.a conftest.b 2>&5 || hard_links=no
 ln conftest.a conftest.b 2>/dev/null && hard_links=no
 AC_MSG_RESULT([$hard_links])
 if test no = "$hard_links"; then
   AC_MSG_WARN(['$CC' does not support '-c -o', so 'make -j' may be unsafe])
   need_locks=warn
 fi
else
 need_locks=no
fi
_LT_DECL([], [need_locks], [1], [Must we lock files when doing compilation?])
])# _LT_COMPILER_FILE_LOCKS


# _LT_CHECK_OBJDIR
# ----------------
m4_defun([_LT_CHECK_OBJDIR],
[AC_CACHE_CHECK([for objdir], [lt_cv_objdir],
[rm -f .libs 2>/dev/null
mkdir .libs 2>/dev/null
if test -d .libs; then
 lt_cv_objdir=.libs
else
 # MS-DOS does not allow filenames that begin with a dot.
 lt_cv_objdir=_libs
fi
rmdir .libs 2>/dev/null])
objdir=$lt_cv_objdir
_LT_DECL([], [objdir], [0],
        [The name of the directory that contains temporary libtool files])dnl
m4_pattern_allow([LT_OBJDIR])dnl
AC_DEFINE_UNQUOTED([LT_OBJDIR], "$lt_cv_objdir/",
 [Define to the sub-directory where libtool stores uninstalled libraries.])
])# _LT_CHECK_OBJDIR


# _LT_LINKER_HARDCODE_LIBPATH([TAGNAME])
# --------------------------------------
# Check hardcoding attributes.
m4_defun([_LT_LINKER_HARDCODE_LIBPATH],
[AC_MSG_CHECKING([how to hardcode library paths into programs])
_LT_TAGVAR(hardcode_action, $1)=
if test -n "$_LT_TAGVAR(hardcode_libdir_flag_spec, $1)" ||
  test -n "$_LT_TAGVAR(runpath_var, $1)" ||
  test yes = "$_LT_TAGVAR(hardcode_automatic, $1)"; then

 # We can hardcode non-existent directories.
 if test no != "$_LT_TAGVAR(hardcode_direct, $1)" &&
    # If the only mechanism to avoid hardcoding is shlibpath_var, we
    # have to relink, otherwise we might link with an installed library
    # when we should be linking with a yet-to-be-installed one
    ## test no != "$_LT_TAGVAR(hardcode_shlibpath_var, $1)" &&
    test no != "$_LT_TAGVAR(hardcode_minus_L, $1)"; then
   # Linking always hardcodes the temporary library directory.
   _LT_TAGVAR(hardcode_action, $1)=relink
 else
   # We can link without hardcoding, and we can hardcode nonexisting dirs.
   _LT_TAGVAR(hardcode_action, $1)=immediate
 fi
else
 # We cannot hardcode anything, or else we can only hardcode existing
 # directories.
 _LT_TAGVAR(hardcode_action, $1)=unsupported
fi
AC_MSG_RESULT([$_LT_TAGVAR(hardcode_action, $1)])

if test relink = "$_LT_TAGVAR(hardcode_action, $1)" ||
  test yes = "$_LT_TAGVAR(inherit_rpath, $1)"; then
 # Fast installation is not supported
 enable_fast_install=no
elif test yes = "$shlibpath_overrides_runpath" ||
    test no = "$enable_shared"; then
 # Fast installation is not necessary
 enable_fast_install=needless
fi
_LT_TAGDECL([], [hardcode_action], [0],
   [How to hardcode a shared library path into an executable])
])# _LT_LINKER_HARDCODE_LIBPATH


# _LT_CMD_STRIPLIB
# ----------------
m4_defun([_LT_CMD_STRIPLIB],
[m4_require([_LT_DECL_EGREP])
striplib=
old_striplib=
AC_MSG_CHECKING([whether stripping libraries is possible])
if test -n "$STRIP" && $STRIP -V 2>&1 | $GREP "GNU strip" >/dev/null; then
 test -z "$old_striplib" && old_striplib="$STRIP --strip-debug"
 test -z "$striplib" && striplib="$STRIP --strip-unneeded"
 AC_MSG_RESULT([yes])
else
# FIXME - insert some real tests, host_os isn't really good enough
 case $host_os in
 darwin*)
```

```
   if test -n "$STRIP"; then
     striplib="$STRIP -x"
     old_striplib="$STRIP -S"
     AC_MSG_RESULT([yes])
   else
     AC_MSG_RESULT([no])
   fi
   ;;
 *)
   AC_MSG_RESULT([no])
   ;;
 esac
fi
_LT_DECL([], [old_striplib], [1], [Commands to strip libraries])
_LT_DECL([], [striplib], [1])
])# _LT_CMD_STRIPLIB


# _LT_PREPARE_MUNGE_PATH_LIST
# ---------------------------
# Make sure func_munge_path_list() is defined correctly.
m4_defun([_LT_PREPARE_MUNGE_PATH_LIST],
[[# func_munge_path_list VARIABLE PATH
# -----------------------------------
# VARIABLE is name of variable containing _space_ separated list of
# directories to be munged by the contents of PATH, which is string
# having a format:
# "DIR[:DIR]:"
#       string "DIR[ DIR]" will be prepended to VARIABLE
# ":DIR[:DIR]"
#       string "DIR[ DIR]" will be appended to VARIABLE
# "DIRP[:DIRP]::[DIRA:]DIRA"
#       string "DIRP[ DIRP]" will be prepended to VARIABLE and string
#       "DIRA[ DIRA]" will be appended to VARIABLE
# "DIR[:DIR]"
#       VARIABLE will be replaced by "DIR[ DIR]"
func_munge_path_list ()
{
   case x@S|@2 in
   x)
       ;;
   *:)
       eval @S|@1=\"`$ECHO @S|@2 | $SED 's/:/ /g'` \@S|@@S|@1\"
       ;;
   x:*)
       eval @S|@1=\"\@S|@@S|@1 `$ECHO @S|@2 | $SED 's/:/ /g'`\"
       ;;
   *::*)
```

```
      eval @S|@1=\"\@S|@@S|@1\ `$ECHO @S|@2 | $SED -e 's/.*:://' -e 's/:/ /g'`\"
      eval @S|@1=\"`$ECHO @S|@2 | $SED -e 's/::.*//' -e 's/:/ /g'`\ \@S|@@S|@1\"
      ;;
    *)
      eval @S|@1=\"`$ECHO @S|@2 | $SED 's/:/ /g'`\"
      ;;
    esac
}
]])# _LT_PREPARE_PATH_LIST


# _LT_SYS_DYNAMIC_LINKER([TAG])
# -----------------------------
# PORTME Fill in your ld.so characteristics
m4_defun([_LT_SYS_DYNAMIC_LINKER],
[AC_REQUIRE([AC_CANONICAL_HOST])dnl
m4_require([_LT_DECL_EGREP])dnl
m4_require([_LT_FILEUTILS_DEFAULTS])dnl
m4_require([_LT_DECL_OBJDUMP])dnl
m4_require([_LT_DECL_SED])dnl
m4_require([_LT_CHECK_SHELL_FEATURES])dnl
m4_require([_LT_PREPARE_MUNGE_PATH_LIST])dnl
AC_MSG_CHECKING([dynamic linker characteristics])
m4_if([$1],
 [], [
if test yes = "$GCC"; then
 case $host_os in
   darwin*) lt_awk_arg='/^libraries:/,/LR/' ;;
   *) lt_awk_arg='/^libraries:/' ;;
 esac
 case $host_os in
   mingw* | cegcc*) lt_sed_strip_eq='s|=\([[A-Za-z]]:\)|\1|g' ;;
   *) lt_sed_strip_eq='s|=/|/|g' ;;
 esac
 lt_search_path_spec=`$CC -print-search-dirs | awk $lt_awk_arg | $SED -e "s/^libraries://" -e $lt_sed_strip_eq`
 case $lt_search_path_spec in
 *\;*)
   # if the path contains ";" then we assume it to be the separator
   # otherwise default to the standard path separator (i.e. ":") - it is
   # assumed that no part of a normal pathname contains ";" but that should
   # okay in the real world where ";" in dirpaths is itself problematic.
   lt_search_path_spec=`$ECHO "$lt_search_path_spec" | $SED 's/;/ /g'`
   ;;
 *)
   lt_search_path_spec=`$ECHO "$lt_search_path_spec" | $SED "s/$PATH_SEPARATOR/ /g"`
   ;;
 esac
 # Ok, now we have the path, separated by spaces, we can step through it
 # and add multilib dir if necessary...
 lt_tmp_lt_search_path_spec=
 lt_multi_os_dir=/`$CC $CPPFLAGS $CFLAGS $LDFLAGS -print-multi-os-directory 2>/dev/null`
 # ...but if some path component already ends with the multilib dir we assume
 # that all is fine and trust -print-search-dirs as is (GCC 4.2? or newer).
 case "$lt_multi_os_dir; $lt_search_path_spec " in
 "/; "* | "/.; "* | "/./; "* | *"$lt_multi_os_dir "* | *"$lt_multi_os_dir/ "*)
   lt_multi_os_dir=
   ;;
 esac
 for lt_sys_path in $lt_search_path_spec; do
   if test -d "$lt_sys_path$lt_multi_os_dir"; then
     lt_tmp_lt_search_path_spec="$lt_tmp_lt_search_path_spec $lt_sys_path$lt_multi_os_dir"
   elif test -n "$lt_multi_os_dir"; then
     test -d "$lt_sys_path" && \
 lt_tmp_lt_search_path_spec="$lt_tmp_lt_search_path_spec $lt_sys_path"
   fi
 done
 lt_search_path_spec=`$ECHO "$lt_tmp_lt_search_path_spec" | awk '
BEGIN {RS = " "; FS = "/|\n";} {
 lt_foo = "";
 lt_count = 0;
 for (lt_i = NF; lt_i > 0; lt_i--) {
   if ($lt_i != "" && $lt_i != ".") {
     if ($lt_i == "..") {
       lt_count++;
     } else {
       if (lt_count == 0) {
         lt_foo = "/" $lt_i lt_foo;
       } else {
         lt_count--;
       }
     }
   }
 }
 if (lt_foo != "") { lt_freq[[lt_foo]]++; }
 if (lt_freq[[lt_foo]] == 1) { print lt_foo; }
}'`
 # AWK program above erroneously prepends '/' to C:/dos/paths
 # for these hosts.
 case $host_os in
   mingw* | cegcc*) lt_search_path_spec=`$ECHO "$lt_search_path_spec" |\
     $SED 's|/\([[A-Za-z]]:\)|\1|g'` ;;
 esac
 sys_lib_search_path_spec=`$ECHO "$lt_search_path_spec" | $lt_NL2SP`
else
 sys_lib_search_path_spec="/lib /usr/lib /usr/local/lib"
fi])
```

```
library_names_spec=
libname_spec='lib$name'
soname_spec=
shrext_cmds=.so
postinstall_cmds=
postuninstall_cmds=
finish_cmds=
finish_eval=
shlibpath_var=
shlibpath_overrides_runpath=unknown
version_type=none
dynamic_linker="$host_os ld.so"
sys_lib_dlsearch_path_spec="/lib /usr/lib"
need_lib_prefix=unknown
hardcode_into_libs=no

# when you set need_version to no, make sure it does not cause -set_version
# flags to be left without arguments
need_version=unknown

AC_ARG_VAR([LT_SYS_LIBRARY_PATH],
[User-defined run-time library search path.])

case $host_os in
aix3*)
 version_type=linux # correct to gnu/linux during the next big refactor
 library_names_spec='$libname$release$shared_ext$versuffix $libname.a'
 shlibpath_var=LIBPATH

 # AIX 3 has no versioning support, so we append a major version to the name.
 soname_spec='$libname$release$shared_ext$major'
 ;;

aix[[4-9]]*)
 version_type=linux # correct to gnu/linux during the next big refactor
 need_lib_prefix=no
 need_version=no
 hardcode_into_libs=yes
 if test ia64 = "$host_cpu"; then
   # AIX 5 supports IA64
   library_names_spec='$libname$release$shared_ext$major $libname$release$shared_ext$versuffix
$libname$shared_ext'
   shlibpath_var=LD_LIBRARY_PATH
 else
   # With GCC up to 2.95.x, collect2 would create an import file
   # for dependence libraries.  The import file would start with
   # the line '#! .'.  This would cause the generated library to
   # depend on '.', always an invalid library.  This was fixed in
```

```
  # development snapshots of GCC prior to 3.0.
  case $host_os in
    aix4 | aix4.[[01]] | aix4.[[01]].*)
    if { echo '#if __GNUC__ > 2 || (__GNUC__ == 2 && __GNUC_MINOR__ >= 97)'
   echo ' yes '
   echo '#endif'; } | $CC -E - | $GREP yes > /dev/null; then
 :
    else
 can_build_shared=no
    fi
    ;;
  esac
  # Using Import Files as archive members, it is possible to support
  # filename-based versioning of shared library archives on AIX. While
  # this would work for both with and without runtime linking, it will
  # prevent static linking of such archives. So we do filename-based
  # shared library versioning with .so extension only, which is used
  # when both runtime linking and shared linking is enabled.
  # Unfortunately, runtime linking may impact performance, so we do
  # not want this to be the default eventually. Also, we use the
  # versioned .so libs for executables only if there is the -brtl
  # linker flag in LDFLAGS as well, or --with-aix-soname=svr4 only.
  # To allow for filename-based versioning support, we need to create
  # libNAME.so.V as an archive file, containing:
  # *) an Import File, referring to the versioned filename of the
  #    archive as well as the shared archive member, telling the
  #    bitwidth (32 or 64) of that shared object, and providing the
  #    list of exported symbols of that shared object, eventually
  #    decorated with the 'weak' keyword
  # *) the shared object with the F_LOADONLY flag set, to really avoid
  #    it being seen by the linker.
  # At run time we better use the real file rather than another symlink,
  # but for link time we create the symlink libNAME.so -> libNAME.so.V

  case $with_aix_soname,$aix_use_runtimelinking in
  # AIX (on Power*) has no versioning support, so currently we cannot hardcode correct
  # soname into executable. Probably we can add versioning support to
  # collect2, so additional links can be useful in future.
  aix,yes) # traditional libtool
    dynamic_linker='AIX unversionable lib.so'
    # If using run time linking (on AIX 4.2 or later) use lib<name>.so
    # instead of lib<name>.a to let people know that these are not
    # typical AIX shared libraries.
    library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
    ;;
  aix,no) # traditional AIX only
    dynamic_linker='AIX lib.a[(]lib.so.V[)]'
    # We preserve .a as extension for shared libraries through AIX4.2
    # and later when we are not doing run time linking.
    library_names_spec='$libname$release.a $libname.a'
    soname_spec='$libname$release$shared_ext$major'
    ;;
  svr4,*) # full svr4 only
    dynamic_linker="AIX lib.so.V[(]$shared_archive_member_spec.o[)]"
    library_names_spec='$libname$release$shared_ext$major $libname$shared_ext'
    # We do not specify a path in Import Files, so LIBPATH fires.
    shlibpath_overrides_runpath=yes
    ;;
  *,yes) # both, prefer svr4
    dynamic_linker="AIX lib.so.V[(]$shared_archive_member_spec.o[)], lib.a[(]lib.so.V[)]"
    library_names_spec='$libname$release$shared_ext$major $libname$shared_ext'
    # unpreferred sharedlib libNAME.a needs extra handling
    postinstall_cmds='test -n "$linkname" || linkname="$realname"~func_stripname "" ".so"
"$linkname"~$install_shared_prog "$dir/$func_stripname_result.$libext"
"$destdir/$func_stripname_result.$libext"~test -z "$tstripme" || test -z "$striplib" || $striplib
"$destdir/$func_stripname_result.$libext"'
    postuninstall_cmds='for n in $library_names $old_library; do :; done~func_stripname "" ".so" "$n"~test
"$func_stripname_result" = "$n" || func_append rmfiles " $odir/$func_stripname_result.$libext"'
    # We do not specify a path in Import Files, so LIBPATH fires.
    shlibpath_overrides_runpath=yes
    ;;
  *,no) # both, prefer aix
    dynamic_linker="AIX lib.a[(]lib.so.V[)], lib.so.V[(]$shared_archive_member_spec.o[)]"
    library_names_spec='$libname$release.a $libname.a'
    soname_spec='$libname$release$shared_ext$major'
    # unpreferred sharedlib libNAME.so.V and symlink libNAME.so need extra handling
    postinstall_cmds='test -z "$dlname" || $install_shared_prog $dir/$dlname $destdir/$dlname~test -z "$tstripme" ||
test -z "$striplib" || $striplib $destdir/$dlname~test -n "$linkname" || linkname=$realname~func_stripname "" ".a"
"$linkname"~(cd "$destdir" && $LN_S -f $dlname $func_stripname_result.so)'
    postuninstall_cmds='test -z "$dlname" || func_append rmfiles " $odir/$dlname"~for n in $old_library
$library_names; do :; done~func_stripname "" ".a" "$n"~func_append rmfiles " $odir/$func_stripname_result.so"'
    ;;
  esac
  shlibpath_var=LIBPATH
 fi
 ;;

amigaos*)
 case $host_cpu in
 powerpc)
  # Since July 2007 AmigaOS4 officially supports .so libraries.
  # When compiling the executable, add -use-dynld -Lsobjs: to the compileline.
  library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
  ;;
 m68k)
   library_names_spec='$libname.ixlibrary $libname.a'
   # Create ${libname}_ixlibrary.a entries in /sys/libs.
   finish_eval='for lib in `ls $libdir/*.ixlibrary 2>/dev/null`; do libname=`func_echo_all "$lib" | $SED
'\''s%^.*/\([[^/]]*\)\.ixlibrary$%\1%'\''`; $RM /sys/libs/${libname}_ixlibrary.a; $show "cd /sys/libs && $LN_S $lib
${libname}_ixlibrary.a"; cd /sys/libs && $LN_S $lib ${libname}_ixlibrary.a || exit 1; done'
   ;;
 esac
 ;;

beos*)
 library_names_spec='$libname$shared_ext'
 dynamic_linker="$host_os ld.so"
 shlibpath_var=LIBRARY_PATH
 ;;

bsdi[[45]]*)
 version_type=linux # correct to gnu/linux during the next big refactor
 need_version=no
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 soname_spec='$libname$release$shared_ext$major'
 finish_cmds='PATH="\$PATH:/sbin" ldconfig $libdir'
 shlibpath_var=LD_LIBRARY_PATH
 sys_lib_search_path_spec="/shlib /usr/lib /usr/X11/lib /usr/contrib/lib /lib /usr/local/lib"
 sys_lib_dlsearch_path_spec="/shlib /usr/lib /usr/local/lib"
 # the default ld.so.conf also contains /usr/contrib/lib and
 # /usr/X11R6/lib (/usr/X11 is a link to /usr/X11R6), but let us allow
 # libtool to hard-code these into programs
 ;;

cygwin* | mingw* | pw32* | cegcc*)
 version_type=windows
 shrext_cmds=.dll
 need_version=no
 need_lib_prefix=no

 case $GCC,$cc_basename in
 yes,*)
   # gcc
   library_names_spec='$libname.dll.a'
   # DLL is installed to $(libdir)/../bin by postinstall_cmds
   postinstall_cmds='base_file=`basename \$file`~
     dlpath=`$SHELL 2>&1 -c '\''. $dir/'\''\$base_file'\''i; echo \$dlname'\''`~
     dldir=$destdir/`dirname \$dlpath`~
     test -d \$dldir || mkdir -p \$dldir~
     $install_prog $dir/$dlname \$dldir/$dlname~
     chmod a+x \$dldir/$dlname~
```

```
      if test -n '\"$stripme'\" && test -n '\"$striplib'\"; then
        eval '\"$striplib \$dldir/$dlname'\" || exit \$?;
      fi'
    postuninstall_cmds='dldll=`$SHELL 2>&1 -c '\''. $file; echo \$dlname'\''`~
      dlpath=$dir/\$dldll~
       $RM \$dlpath'
    shlibpath_overrides_runpath=yes

    case $host_os in
    cygwin*)
      # Cygwin DLLs use 'cyg' prefix rather than 'lib'
      soname_spec='`echo $libname | sed -e 's/^lib/cyg/'``echo $release | $SED -e 's/[[.]]/-/g'`$versuffix$shared_ext'
m4_if([$1], [],[
      sys_lib_search_path_spec="$sys_lib_search_path_spec /usr/lib/w32api"])
      ;;
    mingw* | cegcc*)
      # MinGW DLLs use traditional 'lib' prefix
      soname_spec='$libname`echo $release | $SED -e 's/[[.]]/-/g'`$versuffix$shared_ext'
      ;;
    pw32*)
      # pw32 DLLs use 'pw' prefix rather than 'lib'
      library_names_spec='`echo $libname | sed -e 's/^lib/pw/'``echo $release | $SED -e 's/[[.]]/-
/g'`$versuffix$shared_ext'
      ;;
    esac
    dynamic_linker='Win32 ld.exe'
    ;;

  *,cl*)
    # Native MSVC
    libname_spec='$name'
    soname_spec='$libname`echo $release | $SED -e 's/[[.]]/-/g'`$versuffix$shared_ext'
    library_names_spec='$libname.dll.lib'

    case $build_os in
    mingw*)
      sys_lib_search_path_spec=
      lt_save_ifs=$IFS
      IFS=';'
      for lt_path in $LIB
      do
        IFS=$lt_save_ifs
        # Let DOS variable expansion print the short 8.3 style file name.
        lt_path=`cd "$lt_path" 2>/dev/null && cmd //C "for %i in (".") do @echo %~si"`
        sys_lib_search_path_spec="$sys_lib_search_path_spec $lt_path"
      done
      IFS=$lt_save_ifs
      # Convert to MSYS style.
      sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" | sed -e 's|\\\\|/|g' -e 's| \\([[a-zA-Z]]\\):| /\\1|g' -e 's|^ ||'`
      ;;
    cygwin*)
      # Convert to unix form, then to dos form, then back to unix form
      # but this time dos style (no spaces!) so that the unix form looks
      # like /cygdrive/c/PROGRA~1:/cygdr...
      sys_lib_search_path_spec=`cygpath --path --unix "$LIB"`
      sys_lib_search_path_spec=`cygpath --path --dos "$sys_lib_search_path_spec" 2>/dev/null`
      sys_lib_search_path_spec=`cygpath --path --unix "$sys_lib_search_path_spec" | $SED -e "s/$PATH_SEPARATOR/ /g"`
      ;;
    *)
      sys_lib_search_path_spec=$LIB
      if $ECHO "$sys_lib_search_path_spec" | [$GREP ';[c-zC-Z]:/' >/dev/null]; then
        # It is most probably a Windows format PATH.
        sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" | $SED -e 's/;/ /g'`
      else
        sys_lib_search_path_spec=`$ECHO "$sys_lib_search_path_spec" | $SED -e "s/$PATH_SEPARATOR/ /g"`
      fi
      # FIXME: find the short name or the path components, as spaces are
      # common. (e.g. "Program Files" -> "PROGRA~1")
      ;;
    esac

    # DLL is installed to $(libdir)/../bin by postinstall_cmds
    postinstall_cmds='base_file=`basename \$file`~
      dlpath=`$SHELL 2>&1 -c '\''. $dir/'\''\$base_file'\''i; echo \$dlname'\''`~
      dldir=$destdir/`dirname \$dlpath`~
      test -d \$dldir || mkdir -p \$dldir~
      $install_prog $dir/$dlname \$dldir/$dlname'
    postuninstall_cmds='dldll=`$SHELL 2>&1 -c '\''. $file; echo \$dlname'\''`~
      dlpath=$dir/\$dldll~
       $RM \$dlpath'
    shlibpath_overrides_runpath=yes
    dynamic_linker='Win32 link.exe'
    ;;

  *)
    # Assume MSVC wrapper
    library_names_spec='$libname`echo $release | $SED -e 's/[[.]]/-/g'`$versuffix$shared_ext $libname.lib'
    dynamic_linker='Win32 ld.exe'
    ;;
  esac
  # FIXME: first we should search . and the directory the executable is in
  shlibpath_var=PATH
  ;;
```

```
darwin* | rhapsody*)
 dynamic_linker="$host_os dyld"
 version_type=darwin
 need_lib_prefix=no
 need_version=no
 library_names_spec='$libname$release$major$shared_ext $libname$shared_ext'
 soname_spec='$libname$release$major$shared_ext'
 shlibpath_overrides_runpath=yes
 shlibpath_var=DYLD_LIBRARY_PATH
 shrext_cmds='`test .$module = .yes && echo .so || echo .dylib`'
m4_if([$1], [],[
 sys_lib_search_path_spec="$sys_lib_search_path_spec /usr/local/lib"])
 sys_lib_dlsearch_path_spec='/usr/local/lib /lib /usr/lib'
 ;;

dgux*)
 version_type=linux # correct to gnu/linux during the next big refactor
 need_lib_prefix=no
 need_version=no
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 soname_spec='$libname$release$shared_ext$major'
 shlibpath_var=LD_LIBRARY_PATH
 ;;

freebsd* | dragonfly*)
 # DragonFly does not have aout.  When/if they implement a new
 # versioning mechanism, adjust this.
 if test -x /usr/bin/objformat; then
   objformat=`/usr/bin/objformat`
 else
   case $host_os in
   freebsd[[23]].*) objformat=aout ;;
   *) objformat=elf ;;
   esac
 fi
 version_type=freebsd-$objformat
 case $version_type in
   freebsd-elf*)
     library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
     soname_spec='$libname$release$shared_ext$major'
     need_version=no
     need_lib_prefix=no
     ;;
   freebsd-*)
     library_names_spec='$libname$release$shared_ext$versuffix $libname$shared_ext$versuffix'
     need_version=yes
```

```
    ;;
  esac
  shlibpath_var=LD_LIBRARY_PATH
  case $host_os in
  freebsd2.*)
    shlibpath_overrides_runpath=yes
    ;;
  freebsd3.[[01]]* | freebsdelf3.[[01]]*)
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    ;;
  freebsd3.[[2-9]]* | freebsdelf3.[[2-9]]* | \
  freebsd4.[[0-5]] | freebsdelf4.[[0-5]] | freebsd4.1.1 | freebsdelf4.1.1)
    shlibpath_overrides_runpath=no
    hardcode_into_libs=yes
    ;;
  *) # from 4.6 on, and DragonFly
    shlibpath_overrides_runpath=yes
    hardcode_into_libs=yes
    ;;
  esac
  ;;

haiku*)
  version_type=linux # correct to gnu/linux during the next big refactor
  need_lib_prefix=no
  need_version=no
  dynamic_linker="$host_os runtime_loader"
  library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
  soname_spec='$libname$release$shared_ext$major'
  shlibpath_var=LIBRARY_PATH
  shlibpath_overrides_runpath=no
  sys_lib_dlsearch_path_spec='/boot/home/config/lib /boot/common/lib /boot/system/lib'
  hardcode_into_libs=yes
  ;;

hpux9* | hpux10* | hpux11*)
  # Give a soname corresponding to the major version so that dld.sl refuses to
  # link against other versions.
  version_type=sunos
  need_lib_prefix=no
  need_version=no
  case $host_cpu in
  ia64*)
    shrext_cmds='.so'
    hardcode_into_libs=yes
    dynamic_linker="$host_os dld.so"
    shlibpath_var=LD_LIBRARY_PATH
    shlibpath_overrides_runpath=yes # Unless +noenvvar is specified.
    library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
    soname_spec='$libname$release$shared_ext$major'
    if test 32 = "$HPUX_IA64_MODE"; then
     sys_lib_search_path_spec="/usr/lib/hpux32 /usr/local/lib/hpux32 /usr/local/lib"
     sys_lib_dlsearch_path_spec=/usr/lib/hpux32
    else
     sys_lib_search_path_spec="/usr/lib/hpux64 /usr/local/lib/hpux64"
     sys_lib_dlsearch_path_spec=/usr/lib/hpux64
    fi
    ;;
  hppa*64*)
    shrext_cmds='.sl'
    hardcode_into_libs=yes
    dynamic_linker="$host_os dld.sl"
    shlibpath_var=LD_LIBRARY_PATH # How should we handle SHLIB_PATH
    shlibpath_overrides_runpath=yes # Unless +noenvvar is specified.
    library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
    soname_spec='$libname$release$shared_ext$major'
    sys_lib_search_path_spec="/usr/lib/pa20_64 /usr/ccs/lib/pa20_64"
    sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
    ;;
  *)
    shrext_cmds='.sl'
    dynamic_linker="$host_os dld.sl"
    shlibpath_var=SHLIB_PATH
    shlibpath_overrides_runpath=no # +s is required to enable SHLIB_PATH
    library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
    soname_spec='$libname$release$shared_ext$major'
    ;;
  esac
  # HP-UX runs *really* slowly unless shared libraries are mode 555, ...
  postinstall_cmds='chmod 555 $lib'
  # or fails outright, so override atomically:
  install_override_mode=555
  ;;

interix[[3-9]]*)
  version_type=linux # correct to gnu/linux during the next big refactor
  need_lib_prefix=no
  need_version=no
  library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
  soname_spec='$libname$release$shared_ext$major'
  dynamic_linker='Interix 3.x ld.so.1 (PE, like ELF)'
  shlibpath_var=LD_LIBRARY_PATH
  shlibpath_overrides_runpath=no
  hardcode_into_libs=yes
  ;;

irix5* | irix6* | nonstopux*)
  case $host_os in
    nonstopux*) version_type=nonstopux ;;
    *)
  if test yes = "$lt_cv_prog_gnu_ld"; then
   version_type=linux # correct to gnu/linux during the next big refactor
  else
   version_type=irix
  fi ;;
  esac
  need_lib_prefix=no
  need_version=no
  soname_spec='$libname$release$shared_ext$major'
  library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$release$shared_ext $libname$shared_ext'
  case $host_os in
  irix5* | nonstopux*)
    libsuff= shlibsuff=
    ;;
  *)
    case $LD in # libtool.m4 will add one of these switches to LD
    *-32|*"-32 "|*-melf32bsmip|*"-melf32bsmip ")
     libsuff= shlibsuff= libmagic=32-bit;;
    *-n32|*"-n32 "|*-melf32bmipn32|*"-melf32bmipn32 ")
     libsuff=32 shlibsuff=N32 libmagic=N32;;
    *-64|*"-64 "|*-melf64bmip|*"-melf64bmip ")
     libsuff=64 shlibsuff=64 libmagic=64-bit;;
    *) libsuff= shlibsuff= libmagic=never-match;;
    esac
    ;;
  esac
  shlibpath_var=LD_LIBRARY${shlibsuff}_PATH
  shlibpath_overrides_runpath=no
  sys_lib_search_path_spec="/usr/lib$libsuff /lib$libsuff /usr/local/lib$libsuff"
  sys_lib_dlsearch_path_spec="/usr/lib$libsuff /lib$libsuff"
  hardcode_into_libs=yes
  ;;

# No shared lib support for Linux oldld, aout, or coff.
linux*oldld* | linux*aout* | linux*coff*)
  dynamic_linker=no
  ;;
```

```
linux*android*)
 version_type=none # Android doesn't support versioned libraries.
 need_lib_prefix=no
 need_version=no
 library_names_spec='$libname$release$shared_ext'
 soname_spec='$libname$release$shared_ext'
 finish_cmds=
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=yes

 # This implies no fast_install, which is unacceptable.
 # Some rework will be needed to allow for fast_install
 # before this can be enabled.
 hardcode_into_libs=yes

 dynamic_linker='Android linker'
 # Don't embed -rpath directories since the linker doesn't support them.
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
 ;;

# This must be glibc/ELF.
linux* | k*bsd*-gnu | kopensolaris*-gnu | gnu*)
 version_type=linux # correct to gnu/linux during the next big refactor
 need_lib_prefix=no
 need_version=no
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 soname_spec='$libname$release$shared_ext$major'
 finish_cmds='PATH="\$PATH:/sbin" ldconfig -n $libdir'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=no

 # Some binutils ld are patched to set DT_RUNPATH
 AC_CACHE_VAL([lt_cv_shlibpath_overrides_runpath],
   [lt_cv_shlibpath_overrides_runpath=no
   save_LDFLAGS=$LDFLAGS
   save_libdir=$libdir
   eval "libdir=/foo; wl=\"$_LT_TAGVAR(lt_prog_compiler_wl, $1)\"; \
  LDFLAGS=\"\$LDFLAGS $_LT_TAGVAR(hardcode_libdir_flag_spec, $1)\""
   AC_LINK_IFELSE([AC_LANG_PROGRAM([],[])],
     [AS_IF([ ($OBJDUMP -p conftest$ac_exeext) 2>/dev/null | grep "RUNPATH.*$libdir" >/dev/null],
  [lt_cv_shlibpath_overrides_runpath=yes])])
   LDFLAGS=$save_LDFLAGS
   libdir=$save_libdir
   ])
 shlibpath_overrides_runpath=$lt_cv_shlibpath_overrides_runpath
 # This implies no fast_install, which is unacceptable.
 # Some rework will be needed to allow for fast_install
 # before this can be enabled.
 hardcode_into_libs=yes

 # Ideally, we could use ldconfig to report *all* directores which are
 # searched for libraries, however this is still not possible.  Aside from not
 # being certain /sbin/ldconfig is available, command
 # 'ldconfig -N -X -v | grep ^/' on 64bit Fedora does not report /usr/lib64,
 # even though it is searched at run-time.  Try to do the best guess by
 # appending ld.so.conf contents (and includes) to the search path.
 if test -f /etc/ld.so.conf; then
   lt_ld_extra=`awk '/^include / { system(sprintf("cd /etc; cat %s 2>/dev/null", \[$]2)); skip = 1; } { if (!skip) print \[$]0; skip = 0; }' < /etc/ld.so.conf | $SED -e 's/#.*//;/^[  ]*hwcap[  ]/d;s/[:, ]/ /g;s/=[^=]*$//;s/=[^= ]* / /g;s/"//g;/^$/d' | tr '\n' ' '`
   sys_lib_dlsearch_path_spec="/lib /usr/lib $lt_ld_extra"
 fi

 # We used to test for /lib/ld.so.1 and disable shared libraries on
 # powerpc, because MkLinux only supported shared libraries with the
 # GNU dynamic linker.  Since this was broken with cross compilers,
 # most powerpc-linux boxes support dynamic linking these days and
 # people can always --disable-shared, the test was removed, and we
 # assume the GNU/Linux dynamic linker is in use.
 dynamic_linker='GNU/Linux ld.so'
 ;;

netbsdelf*-gnu)
 version_type=linux
 need_lib_prefix=no
 need_version=no
 library_names_spec='${libname}${release}${shared_ext}$versuffix ${libname}${release}${shared_ext}$major ${libname}${shared_ext}'
 soname_spec='${libname}${release}${shared_ext}$major'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=no
 hardcode_into_libs=yes
 dynamic_linker='NetBSD ld.elf_so'
 ;;

netbsd*)
 version_type=sunos
 need_lib_prefix=no
 need_version=no
 if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
   library_names_spec='$libname$release$shared_ext$versuffix $libname$shared_ext$versuffix'
   finish_cmds='PATH="\$PATH:/sbin" ldconfig -m $libdir'
   dynamic_linker='NetBSD (a.out) ld.so'
 else
   library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
   soname_spec='$libname$release$shared_ext$major'
   dynamic_linker='NetBSD ld.elf_so'
 fi
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=yes
 hardcode_into_libs=yes
 ;;

newsos6)
 version_type=linux # correct to gnu/linux during the next big refactor
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=yes
 ;;

*nto* | *qnx*)
 version_type=qnx
 need_lib_prefix=no
 need_version=no
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 soname_spec='$libname$release$shared_ext$major'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=no
 hardcode_into_libs=yes
 dynamic_linker='ldqnx.so'
 ;;

openbsd* | bitrig*)
 version_type=sunos
 sys_lib_dlsearch_path_spec=/usr/lib
 need_lib_prefix=no
 if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`"; then
   need_version=no
 else
   need_version=yes
 fi
 library_names_spec='$libname$release$shared_ext$versuffix $libname$shared_ext$versuffix'
 finish_cmds='PATH="\$PATH:/sbin" ldconfig -m $libdir'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=yes
 ;;

os2*)
```

```
 libname_spec='$name'
 version_type=windows
 shrext_cmds=.dll
 need_version=no
 need_lib_prefix=no
 # OS/2 can only load a DLL with a base name of 8 characters or less.
 soname_spec='`test -n "$os2dllname" && libname="$os2dllname";
  v=$($ECHO $release$versuffix | tr -d .-);
  n=$($ECHO $libname | cut -b -$((8 - ${#v})) | tr . _);
  $ECHO $n$v`$shared_ext'
 library_names_spec='${libname}_dll.$libext'
 dynamic_linker='OS/2 ld.exe'
 shlibpath_var=BEGINLIBPATH
 sys_lib_search_path_spec="/lib /usr/lib /usr/local/lib"
 sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
 postinstall_cmds='base_file=`basename \$file`~
  dlpath=`$SHELL 2>&1 -c '\''. $dir/'\''\$base_file'\''i; $ECHO \$dlname'\''`~
  dldir=$destdir/`dirname \$dlpath`~
  test -d \$dldir || mkdir -p \$dldir~
  $install_prog $dir/$dlname \$dldir/$dlname~
  chmod a+x \$dldir/$dlname~
  if test -n '\''$stripme'\'' && test -n '\''$striplib'\''; then
   eval '\''$striplib \$dldir/$dlname'\'' || exit \$?;
  fi'
 postuninstall_cmds='dldll=`$SHELL 2>&1 -c '\''. $file; $ECHO \$dlname'\''`~
  dlpath=$dir/\$dldll~
  $RM \$dlpath'
 ;;

osf3* | osf4* | osf5*)
 version_type=osf
 need_lib_prefix=no
 need_version=no
 soname_spec='$libname$release$shared_ext$major'
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 shlibpath_var=LD_LIBRARY_PATH
 sys_lib_search_path_spec="/usr/shlib /usr/ccs/lib /usr/lib/cmplrs/cc /usr/lib /usr/local/lib /var/shlib"
 sys_lib_dlsearch_path_spec=$sys_lib_search_path_spec
 ;;

rdos*)
 dynamic_linker=no
 ;;

solaris*)
 version_type=linux # correct to gnu/linux during the next big refactor
 need_lib_prefix=no
 need_version=no
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 soname_spec='$libname$release$shared_ext$major'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=yes
 hardcode_into_libs=yes
 # ldd complains unless libraries are executable
 postinstall_cmds='chmod +x $lib'
 ;;

sunos4*)
 version_type=sunos
 library_names_spec='$libname$release$shared_ext$versuffix $libname$shared_ext$versuffix'
 finish_cmds='PATH="\$PATH:/usr/etc" ldconfig $libdir'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=yes
 if test yes = "$with_gnu_ld"; then
   need_lib_prefix=no
 fi
 need_version=yes
 ;;

sysv4 | sysv4.3*)
 version_type=linux # correct to gnu/linux during the next big refactor
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 soname_spec='$libname$release$shared_ext$major'
 shlibpath_var=LD_LIBRARY_PATH
 case $host_vendor in
   sni)
     shlibpath_overrides_runpath=no
     need_lib_prefix=no
     runpath_var=LD_RUN_PATH
     ;;
   siemens)
     need_lib_prefix=no
     ;;
   motorola)
     need_lib_prefix=no
     need_version=no
     shlibpath_overrides_runpath=no
     sys_lib_search_path_spec='/lib /usr/lib /usr/ccs/lib'
     ;;
 esac
 ;;

sysv4*MP*)
```

```
 if test -d /usr/nec; then
   version_type=linux # correct to gnu/linux during the next big refactor
   library_names_spec='$libname$shared_ext.$versuffix $libname$shared_ext.$major $libname$shared_ext'
   soname_spec='$libname$shared_ext.$major'
   shlibpath_var=LD_LIBRARY_PATH
 fi
 ;;

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX* | sysv4*uw2*)
 version_type=sco
 need_lib_prefix=no
 need_version=no
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext $libname$shared_ext'
 soname_spec='$libname$release$shared_ext$major'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=yes
 hardcode_into_libs=yes
 if test yes = "$with_gnu_ld"; then
   sys_lib_search_path_spec='/usr/local/lib /usr/gnu/lib /usr/ccs/lib /usr/lib /lib'
 else
   sys_lib_search_path_spec='/usr/ccs/lib /usr/lib'
   case $host_os in
     sco3.2v5*)
       sys_lib_search_path_spec="$sys_lib_search_path_spec /lib"
 ;;
   esac
 fi
 sys_lib_dlsearch_path_spec='/usr/lib'
 ;;

tpf*)
 # TPF is a cross-target only.  Preferred cross-host = GNU/Linux.
 version_type=linux # correct to gnu/linux during the next big refactor
 need_lib_prefix=no
 need_version=no
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 shlibpath_var=LD_LIBRARY_PATH
 shlibpath_overrides_runpath=no
 hardcode_into_libs=yes
 ;;

uts4*)
 version_type=linux # correct to gnu/linux during the next big refactor
 library_names_spec='$libname$release$shared_ext$versuffix $libname$release$shared_ext$major
$libname$shared_ext'
 soname_spec='$libname$release$shared_ext$major'
 shlibpath_var=LD_LIBRARY_PATH
 ;;

*)
 dynamic_linker=no
 ;;
esac
AC_MSG_RESULT([$dynamic_linker])
test no = "$dynamic_linker" && can_build_shared=no

variables_saved_for_relink="PATH $shlibpath_var $runpath_var"
if test yes = "$GCC"; then
 variables_saved_for_relink="$variables_saved_for_relink GCC_EXEC_PREFIX COMPILER_PATH
LIBRARY_PATH"
fi

if test set = "${lt_cv_sys_lib_search_path_spec+set}"; then
 sys_lib_search_path_spec=$lt_cv_sys_lib_search_path_spec
fi

if test set = "${lt_cv_sys_lib_dlsearch_path_spec+set}"; then
 sys_lib_dlsearch_path_spec=$lt_cv_sys_lib_dlsearch_path_spec
fi

# remember unaugmented sys_lib_dlsearch_path content for libtool script decls...
configure_time_dlsearch_path=$sys_lib_dlsearch_path_spec

# ... but it needs LT_SYS_LIBRARY_PATH munging for other configure-time code
func_munge_path_list sys_lib_dlsearch_path_spec "$LT_SYS_LIBRARY_PATH"

# to be used as default LT_SYS_LIBRARY_PATH value in generated libtool
configure_time_lt_sys_library_path=$LT_SYS_LIBRARY_PATH

_LT_DECL([], [variables_saved_for_relink], [1],
   [Variables whose values should be saved in libtool wrapper scripts and
   restored at link time])
_LT_DECL([], [need_lib_prefix], [0],
   [Do we need the "lib" prefix for modules?])
_LT_DECL([], [need_version], [0], [Do we need a version for libraries?])
_LT_DECL([], [version_type], [0], [Library versioning type])
_LT_DECL([], [runpath_var], [0],  [Shared library runtime path variable])
_LT_DECL([], [shlibpath_var], [0],[Shared library path variable])
_LT_DECL([], [shlibpath_overrides_runpath], [0],
   [Is shlibpath searched before the hard-coded library search path?])
_LT_DECL([], [libname_spec], [1], [Format of library name prefix])
_LT_DECL([], [library_names_spec], [1],
   [[List of archive names.  First name is the real one, the rest are links.
   The last name is the one that the linker finds with -lNAME]])
_LT_DECL([], [soname_spec], [1],
   [[The coded name of the library, if different from the real name]])
_LT_DECL([], [install_override_mode], [1],
   [Permission mode override for installation of shared libraries])
_LT_DECL([], [postinstall_cmds], [2],
   [Command to use after installation of a shared archive])
_LT_DECL([], [postuninstall_cmds], [2],
   [Command to use after uninstallation of a shared archive])
_LT_DECL([], [finish_cmds], [2],
   [Commands used to finish a libtool library installation in a directory])
_LT_DECL([], [finish_eval], [1],
   [[As "finish_cmds", except a single script fragment to be evaled but
   not shown]])
_LT_DECL([], [hardcode_into_libs], [0],
   [Whether we should hardcode library paths into libraries])
_LT_DECL([], [sys_lib_search_path_spec], [2],
   [Compile-time system search path for libraries])
_LT_DECL([sys_lib_dlsearch_path_spec], [configure_time_dlsearch_path], [2],
   [Detected run-time system search path for libraries])
_LT_DECL([], [configure_time_lt_sys_library_path], [2],
   [Explicit LT_SYS_LIBRARY_PATH set during ./configure time])
])# _LT_SYS_DYNAMIC_LINKER


# _LT_PATH_TOOL_PREFIX(TOOL)
# --------------------------
# find a file program that can recognize shared library
AC_DEFUN([_LT_PATH_TOOL_PREFIX],
[m4_require([_LT_DECL_EGREP])dnl
AC_MSG_CHECKING([for $1])
AC_CACHE_VAL(lt_cv_path_MAGIC_CMD,
[case $MAGIC_CMD in
[[\\/*] |  ?:[\\/]*])
 lt_cv_path_MAGIC_CMD=$MAGIC_CMD # Let the user override the test with a path.
 ;;
*)
 lt_save_MAGIC_CMD=$MAGIC_CMD
 lt_save_ifs=$IFS; IFS=$PATH_SEPARATOR
dnl $ac_dummy forces splitting on constant user-supplied paths.
dnl POSIX.2 word splitting is done only on the output of word expansions,
dnl not every word.  This closes a longstanding sh security hole.
 ac_dummy="m4_if([$2], , $PATH, [$2])"
 for ac_dir in $ac_dummy; do
   IFS=$lt_save_ifs
   test -z "$ac_dir" && ac_dir=.
   if test -f "$ac_dir/$1"; then
     lt_cv_path_MAGIC_CMD=$ac_dir/"$1"
     if test -n "$file_magic_test_file"; then
 case $deplibs_check_method in
 "file_magic "*)
   file_magic_regex=`expr "$deplibs_check_method" : "file_magic \(.*\)"`
   MAGIC_CMD=$lt_cv_path_MAGIC_CMD
   if eval $file_magic_cmd \$file_magic_test_file 2> /dev/null |
     $EGREP "$file_magic_regex" > /dev/null; then
     :
   else
     cat <<_LT_EOF 1>&2

*** Warning: the command libtool uses to detect shared libraries,
*** $file_magic_cmd, produces output that libtool cannot recognize.
*** The result is that libtool may fail to recognize shared libraries
*** as such.  This will affect the creation of libtool libraries that
*** depend on shared libraries, but programs linked with such libtool
*** libraries will work regardless of this problem.  Nevertheless, you
*** may want to report the problem to your system manager and/or to
*** bug-libtool@gnu.org

_LT_EOF
   fi ;;
 esac
     fi
     break
   fi
 done
 IFS=$lt_save_ifs
 MAGIC_CMD=$lt_save_MAGIC_CMD
 ;;
esac])
MAGIC_CMD=$lt_cv_path_MAGIC_CMD
if test -n "$MAGIC_CMD"; then
 AC_MSG_RESULT($MAGIC_CMD)
else
 AC_MSG_RESULT(no)
fi
_LT_DECL([], [MAGIC_CMD], [0],
  [Used to examine libraries when file_magic_cmd begins with "file"])dnl
])# _LT_PATH_TOOL_PREFIX

# Old name:
AU_ALIAS([AC_PATH_TOOL_PREFIX], [_LT_PATH_TOOL_PREFIX])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AC_PATH_TOOL_PREFIX], [])



# _LT_PATH_MAGIC
# --------------
# find a file program that can recognize a shared library
```

```
m4_defun([_LT_PATH_MAGIC],
[_LT_PATH_TOOL_PREFIX(${ac_tool_prefix}file, /usr/bin$PATH_SEPARATOR$PATH)
if test -z "$lt_cv_path_MAGIC_CMD"; then
 if test -n "$ac_tool_prefix"; then
   _LT_PATH_TOOL_PREFIX(file, /usr/bin$PATH_SEPARATOR$PATH)
 else
   MAGIC_CMD=:
 fi
fi
])# _LT_PATH_MAGIC


# LT_PATH_LD
# ----------
# find the pathname to the GNU or non-GNU linker
AC_DEFUN([LT_PATH_LD],
[AC_REQUIRE([AC_PROG_CC])dnl
AC_REQUIRE([AC_CANONICAL_HOST])dnl
AC_REQUIRE([AC_CANONICAL_BUILD])dnl
m4_require([_LT_DECL_SED])dnl
m4_require([_LT_DECL_EGREP])dnl
m4_require([_LT_PROG_ECHO_BACKSLASH])dnl

AC_ARG_WITH([gnu-ld],
   [AS_HELP_STRING([--with-gnu-ld],
 [assume the C compiler uses GNU ld @<:@default=no@:>@])],
   [test no = "$withval" || with_gnu_ld=yes],
   [with_gnu_ld=no])dnl

ac_prog=ld
if test yes = "$GCC"; then
 # Check if gcc -print-prog-name=ld gives a path.
 AC_MSG_CHECKING([for ld used by $CC])
 case $host in
 *-*-mingw*)
   # gcc leaves a trailing carriage return, which upsets mingw
   ac_prog=`($CC -print-prog-name=ld) 2>&5 | tr -d '\015'` ;;
 *)
   ac_prog=`($CC -print-prog-name=ld) 2>&5` ;;
 esac
 case $ac_prog in
   # Accept absolute paths.
   [[\\/]]* | ?:[[\\/]]*)
     re_direlt='/[[^/]][[^/]]*/\.\./'
     # Canonicalize the pathname of ld
     ac_prog=`$ECHO "$ac_prog"| $SED 's%\\\\%/%g'`
     while $ECHO "$ac_prog" | $GREP "$re_direlt" > /dev/null 2>&1; do
 ac_prog=`$ECHO $ac_prog| $SED "s%$re_direlt%/%"`
```

```
    done
    test -z "$LD" && LD=$ac_prog
    ;;
 "")
   # If it fails, then pretend we aren't using GCC.
   ac_prog=ld
   ;;
 *)
   # If it is relative, then search for the first ld in PATH.
   with_gnu_ld=unknown
   ;;
 esac
elif test yes = "$with_gnu_ld"; then
 AC_MSG_CHECKING([for GNU ld])
else
 AC_MSG_CHECKING([for non-GNU ld])
fi
AC_CACHE_VAL(lt_cv_path_LD,
[if test -z "$LD"; then
 lt_save_ifs=$IFS; IFS=$PATH_SEPARATOR
 for ac_dir in $PATH; do
   IFS=$lt_save_ifs
   test -z "$ac_dir" && ac_dir=.
   if test -f "$ac_dir/$ac_prog" || test -f "$ac_dir/$ac_prog$ac_exeext"; then
     lt_cv_path_LD=$ac_dir/$ac_prog
     # Check to see if the program is GNU ld.  I'd rather use --version,
     # but apparently some variants of GNU ld only accept -v.
     # Break only if it was the GNU/non-GNU ld that we prefer.
     case `"$lt_cv_path_LD" -v 2>&1 </dev/null` in
     *GNU* | *'with BFD'*)
 test no != "$with_gnu_ld" && break
 ;;
     *)
 test yes != "$with_gnu_ld" && break
 ;;
     esac
   fi
 done
 IFS=$lt_save_ifs
else
 lt_cv_path_LD=$LD # Let the user override the test with a path.
fi])
LD=$lt_cv_path_LD
if test -n "$LD"; then
 AC_MSG_RESULT($LD)
else
 AC_MSG_RESULT(no)
fi
```

```
test -z "$LD" && AC_MSG_ERROR([no acceptable ld found in \$PATH])
_LT_PATH_LD_GNU
AC_SUBST([LD])

_LT_TAGDECL([], [LD], [1], [The linker used to build libraries])
])# LT_PATH_LD

# Old names:
AU_ALIAS([AM_PROG_LD], [LT_PATH_LD])
AU_ALIAS([AC_PROG_LD], [LT_PATH_LD])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AM_PROG_LD], [])
dnl AC_DEFUN([AC_PROG_LD], [])


# _LT_PATH_LD_GNU
#- --------------
m4_defun([_LT_PATH_LD_GNU],
[AC_CACHE_CHECK([if the linker ($LD) is GNU ld], lt_cv_prog_gnu_ld,
[# I'd rather use --version here, but apparently some GNU lds only accept -v.
case `$LD -v 2>&1 </dev/null` in
*GNU* | *'with BFD'*)
 lt_cv_prog_gnu_ld=yes
 ;;
*)
 lt_cv_prog_gnu_ld=no
 ;;
esac])
with_gnu_ld=$lt_cv_prog_gnu_ld
])# _LT_PATH_LD_GNU


# _LT_CMD_RELOAD
# --------------
# find reload flag for linker
#   -- PORTME Some linkers may need a different reload flag.
m4_defun([_LT_CMD_RELOAD],
[AC_CACHE_CHECK([for $LD option to reload object files],
 lt_cv_ld_reload_flag,
 [lt_cv_ld_reload_flag='-r'])
reload_flag=$lt_cv_ld_reload_flag
case $reload_flag in
"" | " "*) ;;
*) reload_flag=" $reload_flag" ;;
esac
reload_cmds='$LD$reload_flag -o $output$reload_objs'
case $host_os in
 cygwin* | mingw* | pw32* | cegcc*)
```

```
    if test yes != "$GCC"; then
      reload_cmds=false
    fi
    ;;
  darwin*)
    if test yes = "$GCC"; then
      reload_cmds='$LTCC $LTCFLAGS -nostdlib $wl-r -o $output$reload_objs'
    else
      reload_cmds='$LD$reload_flag -o $output$reload_objs'
    fi
    ;;
esac
_LT_TAGDECL([], [reload_flag], [1], [How to create reloadable object files])dnl
_LT_TAGDECL([], [reload_cmds], [2])dnl
])# _LT_CMD_RELOAD


# _LT_PATH_DD
# -----------
# find a working dd
m4_defun([_LT_PATH_DD],
[AC_CACHE_CHECK([for a working dd], [ac_cv_path_lt_DD],
[printf 0123456789abcdef0123456789abcdef >conftest.i
cat conftest.i conftest.i >conftest2.i
: ${lt_DD:=$DD}
AC_PATH_PROGS_FEATURE_CHECK([lt_DD], [dd],
[if "$ac_path_lt_DD" bs=32 count=1 <conftest2.i >conftest.out 2>/dev/null; then
  cmp -s conftest.i conftest.out \
  && ac_cv_path_lt_DD="$ac_path_lt_DD" ac_path_lt_DD_found=:
fi])
rm -f conftest.i conftest2.i conftest.out])
])# _LT_PATH_DD


# _LT_CMD_TRUNCATE
# ----------------
# find command to truncate a binary pipe
m4_defun([_LT_CMD_TRUNCATE],
[m4_require([_LT_PATH_DD])
AC_CACHE_CHECK([how to truncate binary pipes], [lt_cv_truncate_bin],
[printf 0123456789abcdef0123456789abcdef >conftest.i
cat conftest.i conftest.i >conftest2.i
lt_cv_truncate_bin=
if "$ac_cv_path_lt_DD" bs=32 count=1 <conftest2.i >conftest.out 2>/dev/null; then
  cmp -s conftest.i conftest.out \
  && lt_cv_truncate_bin="$ac_cv_path_lt_DD bs=4096 count=1"
fi
rm -f conftest.i conftest2.i conftest.out
test -z "$lt_cv_truncate_bin" && lt_cv_truncate_bin="$SED -e 4q"])
_LT_DECL([lt_truncate_bin], [lt_cv_truncate_bin], [1],
 [Command to truncate a binary pipe])
])# _LT_CMD_TRUNCATE


# _LT_CHECK_MAGIC_METHOD
# ----------------------
# how to check for library dependencies
#  -- PORTME fill in with the dynamic library characteristics
m4_defun([_LT_CHECK_MAGIC_METHOD],
[m4_require([_LT_DECL_EGREP])
m4_require([_LT_DECL_OBJDUMP])
AC_CACHE_CHECK([how to recognize dependent libraries],
lt_cv_deplibs_check_method,
[lt_cv_file_magic_cmd='$MAGIC_CMD'
lt_cv_file_magic_test_file=
lt_cv_deplibs_check_method='unknown'
# Need to set the preceding variable on all platforms that support
# interlibrary dependencies.
# 'none' -- dependencies not supported.
# 'unknown' -- same as none, but documents that we really don't know.
# 'pass_all' -- all dependencies passed with no checks.
# 'test_compile' -- check by making test program.
# 'file_magic [[regex]]' -- check by looking for files in library path
# that responds to the $file_magic_cmd with a given extended regex.
# If you have 'file' or equivalent on your system and you're not sure
# whether 'pass_all' will *always* work, you probably want this one.

case $host_os in
aix[[4-9]]*)
 lt_cv_deplibs_check_method=pass_all
 ;;

beos*)
 lt_cv_deplibs_check_method=pass_all
 ;;

bsdi[[45]]*)
 lt_cv_deplibs_check_method='file_magic ELF [[0-9]][[0-9]]*-bit [[ML]]SB (shared object|dynamic lib)'
 lt_cv_file_magic_cmd='/usr/bin/file -L'
 lt_cv_file_magic_test_file=/shlib/libc.so
 ;;

cygwin*)
 # func_win32_libid is a shell function defined in ltmain.sh
 lt_cv_deplibs_check_method='file_magic ^x86 archive import|^x86 DLL'
 lt_cv_file_magic_cmd='func_win32_libid'
 ;;

mingw* | pw32*)
 # Base MSYS/MinGW do not provide the 'file' command needed by
 # func_win32_libid shell function, so use a weaker test based on 'objdump',
 # unless we find 'file', for example because we are cross-compiling.
 if ( file / ) >/dev/null 2>&1; then
   lt_cv_deplibs_check_method='file_magic ^x86 archive import|^x86 DLL'
   lt_cv_file_magic_cmd='func_win32_libid'
 else
   # Keep this pattern in sync with the one in func_win32_libid.
   lt_cv_deplibs_check_method='file_magic file format (pei*-i386(.*architecture: i386)?|pe-arm-wince|pe-x86-64)'
   lt_cv_file_magic_cmd='$OBJDUMP -f'
 fi
 ;;

cegcc*)
 # use the weaker test based on 'objdump'. See mingw*.
 lt_cv_deplibs_check_method='file_magic file format pe-arm-.*little(.*architecture: arm)?'
 lt_cv_file_magic_cmd='$OBJDUMP -f'
 ;;

darwin* | rhapsody*)
 lt_cv_deplibs_check_method=pass_all
 ;;

freebsd* | dragonfly*)
 if echo __ELF__ | $CC -E - | $GREP __ELF__ > /dev/null; then
   case $host_cpu in
   i*86 )
     # Not sure whether the presence of OpenBSD here was a mistake.
     # Let's accept both of them until this is cleared up.
     lt_cv_deplibs_check_method='file_magic (FreeBSD|OpenBSD|DragonFly)/i[[3-9]]86 (compact )?demand paged shared library'
     lt_cv_file_magic_cmd=/usr/bin/file
     lt_cv_file_magic_test_file=`echo /usr/lib/libc.so.*`
     ;;
   esac
 else
   lt_cv_deplibs_check_method=pass_all
 fi
 ;;

haiku*)
 lt_cv_deplibs_check_method=pass_all
 ;;

hpux10.20* | hpux11*)
```

```
 lt_cv_file_magic_cmd=/usr/bin/file
 case $host_cpu in
 ia64*)
   lt_cv_deplibs_check_method='file_magic (s[[0-9]][[0-9]][[0-9]]|ELF-[[0-9]][[0-9]]) shared object file - IA64'
   lt_cv_file_magic_test_file=/usr/lib/hpux32/libc.so
   ;;
 hppa*64*)
   [lt_cv_deplibs_check_method='file_magic (s[0-9][0-9][0-9]|ELF[ -][0-9][0-9])(-bit)?( [LM]SB)? shared object(
file)?[, -]* PA-RISC [0-9]\.[0-9]']
   lt_cv_file_magic_test_file=/usr/lib/pa20_64/libc.sl
   ;;
 *)
   lt_cv_deplibs_check_method='file_magic (s[[0-9]][[0-9]][[0-9]]|PA-RISC[[0-9]]\.[[0-9]]) shared library'
   lt_cv_file_magic_test_file=/usr/lib/libc.sl
   ;;
 esac
 ;;

interix[[3-9]]*)
 # PIC code is broken on Interix 3.x, that's why |\.a not |_pic\.a here
 lt_cv_deplibs_check_method='match_pattern /lib[[^/]]+(\.so|\.a)$'
 ;;

irix5* | irix6* | nonstopux*)
 case $LD in
 *-32|*"-32 ") libmagic=32-bit;;
 *-n32|*"-n32 ") libmagic=N32;;
 *-64|*"-64 ") libmagic=64-bit;;
 *) libmagic=never-match;;
 esac
 lt_cv_deplibs_check_method=pass_all
 ;;

# This must be glibc/ELF.
linux* | k*bsd*-gnu | kopensolaris*-gnu | gnu*)
 lt_cv_deplibs_check_method=pass_all
 ;;

netbsd* | netbsdelf*-gnu)
 if echo __ELF__ | $CC -E - | $GREP __ELF__ > /dev/null; then
   lt_cv_deplibs_check_method='match_pattern /lib[[^/]]+(\.so\.[[0-9]]+\.[[0-9]]+|_pic\.a)$'
 else
   lt_cv_deplibs_check_method='match_pattern /lib[[^/]]+(\.so|_pic\.a)$'
 fi
 ;;

newos6*)
 lt_cv_deplibs_check_method='file_magic ELF [[0-9]][[0-9]]*-bit [[ML]]SB (executable|dynamic lib)'
 lt_cv_file_magic_cmd=/usr/bin/file
 lt_cv_file_magic_test_file=/usr/lib/libnls.so
 ;;

*nto* | *qnx*)
 lt_cv_deplibs_check_method=pass_all
 ;;

openbsd* | bitrig*)
 if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`"; then
   lt_cv_deplibs_check_method='match_pattern /lib[[^/]]+(\.so\.[[0-9]]+\.[[0-9]]+|\.so|_pic\.a)$'
 else
   lt_cv_deplibs_check_method='match_pattern /lib[[^/]]+(\.so\.[[0-9]]+\.[[0-9]]+|_pic\.a)$'
 fi
 ;;

osf3* | osf4* | osf5*)
 lt_cv_deplibs_check_method=pass_all
 ;;

rdos*)
 lt_cv_deplibs_check_method=pass_all
 ;;

solaris*)
 lt_cv_deplibs_check_method=pass_all
 ;;

sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX* | sysv4*uw2*)
 lt_cv_deplibs_check_method=pass_all
 ;;

sysv4 | sysv4.3*)
 case $host_vendor in
 motorola)
   lt_cv_deplibs_check_method='file_magic ELF [[0-9]][[0-9]]*-bit [[ML]]SB (shared object|dynamic lib) M[[0-9]][[0-9]]* Version [[0-9]]'
   lt_cv_file_magic_test_file=`echo /usr/lib/libc.so*`
   ;;
 ncr)
   lt_cv_deplibs_check_method=pass_all
   ;;
 sequent)
   lt_cv_file_magic_cmd='/bin/file'
   lt_cv_deplibs_check_method='file_magic ELF [[0-9]][[0-9]]*-bit [[LM]]SB (shared object|dynamic lib )'
   ;;
 sni)
   lt_cv_file_magic_cmd='/bin/file'
   lt_cv_deplibs_check_method="file_magic ELF [[0-9]][[0-9]]*-bit [[LM]]SB dynamic lib"
   lt_cv_file_magic_test_file=/lib/libc.so
   ;;
  siemens)
   lt_cv_deplibs_check_method=pass_all
   ;;
  pc)
   lt_cv_deplibs_check_method=pass_all
   ;;
  esac
  ;;

tpf*)
  lt_cv_deplibs_check_method=pass_all
  ;;
os2*)
  lt_cv_deplibs_check_method=pass_all
  ;;
esac
])

file_magic_glob=
want_nocaseglob=no
if test "$build" = "$host"; then
  case $host_os in
  mingw* | pw32*)
   if ( shopt | grep nocaseglob ) >/dev/null 2>&1; then
     want_nocaseglob=yes
   else
     file_magic_glob=`echo aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ | $SED -e
"s/\(..\)/s\/[[\1]]\/[[\1]]\/g;/g"`
   fi
   ;;
  esac
fi

file_magic_cmd=$lt_cv_file_magic_cmd
deplibs_check_method=$lt_cv_deplibs_check_method
test -z "$deplibs_check_method" && deplibs_check_method=unknown

_LT_DECL([], [deplibs_check_method], [1],
   [Method to check whether dependent libraries are shared objects])
_LT_DECL([], [file_magic_cmd], [1],
   [Command to use when deplibs_check_method = "file_magic"])
_LT_DECL([], [file_magic_glob], [1],
   [How to find potential files when deplibs_check_method = "file_magic"])
_LT_DECL([], [want_nocaseglob], [1],
   [Find potential files using nocaseglob when deplibs_check_method = "file_magic"])
```

```
])# _LT_CHECK_MAGIC_METHOD


# LT_PATH_NM
# ----------
# find the pathname to a BSD- or MS-compatible name lister
AC_DEFUN([LT_PATH_NM],
[AC_REQUIRE([AC_PROG_CC])dnl
AC_CACHE_CHECK([for BSD- or MS-compatible name lister (nm)], lt_cv_path_NM,
[if test -n "$NM"; then
 # Let the user override the test.
 lt_cv_path_NM=$NM
else
 lt_nm_to_check=${ac_tool_prefix}nm
 if test -n "$ac_tool_prefix" && test "$build" = "$host"; then
   lt_nm_to_check="$lt_nm_to_check nm"
 fi
 for lt_tmp_nm in $lt_nm_to_check; do
   lt_save_ifs=$IFS; IFS=$PATH_SEPARATOR
   for ac_dir in $PATH /usr/ccs/bin/elf /usr/ccs/bin /usr/ucb /bin; do
     IFS=$lt_save_ifs
     test -z "$ac_dir" && ac_dir=.
     tmp_nm=$ac_dir/$lt_tmp_nm
     if test -f "$tmp_nm" || test -f "$tmp_nm$ac_exeext"; then
 # Check to see if the nm accepts a BSD-compat flag.
 # Adding the 'sed 1q' prevents false positives on HP-UX, which says:
 #   nm: unknown option "B" ignored
 # Tru64's nm complains that /dev/null is an invalid object file
 # MSYS converts /dev/null to NUL, MinGW nm treats NUL as empty
 case $build_os in
 mingw*) lt_bad_file=conftest.nm/nofile ;;
 *) lt_bad_file=/dev/null ;;
 esac
 case `"$tmp_nm" -B $lt_bad_file 2>&1 | sed '1q'` in
 *$lt_bad_file* | *'Invalid file or object type'*)
   lt_cv_path_NM="$tmp_nm -B"
   break 2
   ;;
 *)
   case `"$tmp_nm" -p /dev/null 2>&1 | sed '1q'` in
   */dev/null*)
     lt_cv_path_NM="$tmp_nm -p"
     break 2
     ;;
   *)
     lt_cv_path_NM=${lt_cv_path_NM="$tmp_nm"} # keep the first match, but
     continue # so that we can try to find one that supports BSD flags
     ;;
   esac
   ;;
 esac
     fi
   done
   IFS=$lt_save_ifs
 done
 : ${lt_cv_path_NM=no}
fi])
if test no != "$lt_cv_path_NM"; then
 NM=$lt_cv_path_NM
else
 # Didn't find any BSD compatible name lister, look for dumpbin.
 if test -n "$DUMPBIN"; then :
   # Let the user override the test.
 else
   AC_CHECK_TOOLS(DUMPBIN, [dumpbin "link -dump"], :)
   case `$DUMPBIN -symbols -headers /dev/null 2>&1 | sed '1q'` in
   *COFF*)
     DUMPBIN="$DUMPBIN -symbols -headers"
     ;;
   *)
     DUMPBIN=:
     ;;
   esac
 fi
 AC_SUBST([DUMPBIN])
 if test : != "$DUMPBIN"; then
   NM=$DUMPBIN
 fi
fi
test -z "$NM" && NM=nm
AC_SUBST([NM])
_LT_DECL([], [NM], [1], [A BSD- or MS-compatible name lister])dnl

AC_CACHE_CHECK([the name lister ($NM) interface], [lt_cv_nm_interface],
 [lt_cv_nm_interface="BSD nm"
 echo "int some_variable = 0;" > conftest.$ac_ext
 (eval echo "\"\$as_me:$LINENO: $ac_compile\"" >&AS_MESSAGE_LOG_FD)
 (eval "$ac_compile" 2>conftest.err)
 cat conftest.err >&AS_MESSAGE_LOG_FD
 (eval echo "\"\$as_me:$LINENO: $NM \\\"conftest.$ac_objext\\\"\"" >&AS_MESSAGE_LOG_FD)
 (eval "$NM \"conftest.$ac_objext\"" 2>conftest.err > conftest.out)
 cat conftest.err >&AS_MESSAGE_LOG_FD
 (eval echo "\"\$as_me:$LINENO: output\"" >&AS_MESSAGE_LOG_FD)
 cat conftest.out >&AS_MESSAGE_LOG_FD
 if $GREP 'External.*some_variable' conftest.out > /dev/null; then
   lt_cv_nm_interface="MS dumpbin"
 fi
 rm -f conftest*])
])# LT_PATH_NM

# Old names:
AU_ALIAS([AM_PROG_NM], [LT_PATH_NM])
AU_ALIAS([AC_PROG_NM], [LT_PATH_NM])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AM_PROG_NM], [])
dnl AC_DEFUN([AC_PROG_NM], [])

# _LT_CHECK_SHAREDLIB_FROM_LINKLIB
# --------------------------------
# how to determine the name of the shared library
# associated with a specific link library.
#  -- PORTME fill in with the dynamic library characteristics
m4_defun([_LT_CHECK_SHAREDLIB_FROM_LINKLIB],
[m4_require([_LT_DECL_EGREP])
m4_require([_LT_DECL_OBJDUMP])
m4_require([_LT_DECL_DLLTOOL])
AC_CACHE_CHECK([how to associate runtime and link libraries],
lt_cv_sharedlib_from_linklib_cmd,
[lt_cv_sharedlib_from_linklib_cmd='unknown'

case $host_os in
cygwin* | mingw* | pw32* | cegcc*)
 # two different shell functions defined in ltmain.sh;
 # decide which one to use based on capabilities of $DLLTOOL
 case `$DLLTOOL --help 2>&1` in
 *--identify-strict*)
   lt_cv_sharedlib_from_linklib_cmd=func_cygming_dll_for_implib
   ;;
 *)
   lt_cv_sharedlib_from_linklib_cmd=func_cygming_dll_for_implib_fallback
   ;;
 esac
 ;;
*)
 # fallback: assume linklib IS sharedlib
 lt_cv_sharedlib_from_linklib_cmd=$ECHO
 ;;
esac
])
sharedlib_from_linklib_cmd=$lt_cv_sharedlib_from_linklib_cmd
test -z "$sharedlib_from_linklib_cmd" && sharedlib_from_linklib_cmd=$ECHO

_LT_DECL([], [sharedlib_from_linklib_cmd], [1],
   [Command to associate shared and link libraries])
```

```
])# _LT_CHECK_SHAREDLIB_FROM_LINKLIB


# _LT_PATH_MANIFEST_TOOL
# ----------------------
# locate the manifest tool
m4_defun([_LT_PATH_MANIFEST_TOOL],
[AC_CHECK_TOOL(MANIFEST_TOOL, mt, :)
test -z "$MANIFEST_TOOL" && MANIFEST_TOOL=mt
AC_CACHE_CHECK([if $MANIFEST_TOOL is a manifest tool], [lt_cv_path_mainfest_tool],
 [lt_cv_path_mainfest_tool=no
 echo "$as_me:$LINENO: $MANIFEST_TOOL '-?'" >&AS_MESSAGE_LOG_FD
 $MANIFEST_TOOL '-?' 2>conftest.err > conftest.out
 cat conftest.err >&AS_MESSAGE_LOG_FD
 if $GREP 'Manifest Tool' conftest.out > /dev/null; then
   lt_cv_path_mainfest_tool=yes
 fi
 rm -f conftest*])
if test yes != "$lt_cv_path_mainfest_tool"; then
 MANIFEST_TOOL=:
fi
_LT_DECL([], [MANIFEST_TOOL], [1], [Manifest tool])dnl
])# _LT_PATH_MANIFEST_TOOL


# _LT_DLL_DEF_P([FILE])
# ---------------------
# True iff FILE is a Windows DLL '.def' file.
# Keep in sync with func_dll_def_p in the libtool script
AC_DEFUN([_LT_DLL_DEF_P],
[dnl
 test DEF = "`$SED -n dnl
   -e '\''s/^[[  ]]*//'\'' dnl Strip leading whitespace
   -e '\''/^\(;.*\)*$/d'\'' dnl      Delete empty lines and comments
   -e '\''s/^\(EXPORTS\|LIBRARY\)\([[  ]].*\)*$/DEF/p'\'' dnl
   -e q dnl                          Only consider the first "real" line
   $1`" dnl
])# _LT_DLL_DEF_P


# LT_LIB_M
# --------
# check for math library
AC_DEFUN([LT_LIB_M],
[AC_REQUIRE([AC_CANONICAL_HOST])dnl
LIBM=
case $host in
*-*-beos* | *-*-cegcc* | *-*-cygwin* | *-*-haiku* | *-*-pw32* | *-*-darwin*)
```

```
 # These system don't have libm, or don't need it
 ;;
*-ncr-sysv4.3*)
 AC_CHECK_LIB(mw, _mwvalidcheckl, LIBM=-lmw)
 AC_CHECK_LIB(m, cos, LIBM="$LIBM -lm")
 ;;
*)
 AC_CHECK_LIB(m, cos, LIBM=-lm)
 ;;
esac
AC_SUBST([LIBM])
])# LT_LIB_M

# Old name:
AU_ALIAS([AC_CHECK_LIBM], [LT_LIB_M])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([AC_CHECK_LIBM], [])


# _LT_COMPILER_NO_RTTI([TAGNAME])
# -------------------------------
m4_defun([_LT_COMPILER_NO_RTTI],
[m4_require([_LT_TAG_COMPILER])dnl

_LT_TAGVAR(lt_prog_compiler_no_builtin_flag, $1)=

if test yes = "$GCC"; then
 case $cc_basename in
 nvcc*)
   _LT_TAGVAR(lt_prog_compiler_no_builtin_flag, $1)=' -Xcompiler -fno-builtin' ;;
 *)
   _LT_TAGVAR(lt_prog_compiler_no_builtin_flag, $1)=' -fno-builtin' ;;
 esac

 _LT_COMPILER_OPTION([if $compiler supports -fno-rtti -fno-exceptions],
   lt_cv_prog_compiler_rtti_exceptions,
   [-fno-rtti -fno-exceptions], [],
   [_LT_TAGVAR(lt_prog_compiler_no_builtin_flag, $1)="$_LT_TAGVAR(lt_prog_compiler_no_builtin_flag,
$1) -fno-rtti -fno-exceptions"])
fi
_LT_TAGDECL([no_builtin_flag], [lt_prog_compiler_no_builtin_flag], [1],
 [Compiler flag to turn off builtin functions])
])# _LT_COMPILER_NO_RTTI


# _LT_CMD_GLOBAL_SYMBOLS
# ----------------------
m4_defun([_LT_CMD_GLOBAL_SYMBOLS],
[AC_REQUIRE([AC_CANONICAL_HOST])dnl
AC_REQUIRE([AC_PROG_CC])dnl
AC_REQUIRE([AC_PROG_AWK])dnl
AC_REQUIRE([LT_PATH_NM])dnl
AC_REQUIRE([LT_PATH_LD])dnl
m4_require([_LT_DECL_SED])dnl
m4_require([_LT_DECL_EGREP])dnl
m4_require([_LT_TAG_COMPILER])dnl

# Check for command to grab the raw symbol name followed by C symbol from nm.
AC_MSG_CHECKING([command to parse $NM output from $compiler object])
AC_CACHE_VAL([lt_cv_sys_global_symbol_pipe],
[
# These are sane defaults that work on at least a few old systems.
# [They come from Ultrix.  What could be older than Ultrix?!! ;)]

# Character class describing NM global symbol codes.
symcode='[[BCDEGRST]]'

# Regexp to match symbols that can be accessed directly from C.
sympat='\([[_A-Za-z]][[_A-Za-z0-9]]*\)'

# Define system-specific variables.
case $host_os in
aix*)
 symcode='[[BCDT]]'
 ;;
cygwin* | mingw* | pw32* | cegcc*)
 symcode='[[ABCDGISTW]]'
 ;;
hpux*)
 if test ia64 = "$host_cpu"; then
   symcode='[[ABCDEGRST]]'
 fi
 ;;
irix* | nonstopux*)
 symcode='[[BCDEGRST]]'
 ;;
osf*)
 symcode='[[BCDEGQRST]]'
 ;;
solaris*)
 symcode='[[BDRT]]'
 ;;
sco3.2v5*)
 symcode='[[DT]]'
 ;;
sysv4.2uw2*)
```

```
 symcode='[[DT]]'
 ;;
sysv5* | sco5v6* | unixware* | OpenUNIX*)
 symcode='[[ABDT]]'
 ;;
sysv4)
 symcode='[[DFNSTU]]'
 ;;
esac

# If we're using GNU nm, then use its standard symbol codes.
case `$NM -V 2>&1` in
*GNU* | *'with BFD'*)
 symcode='[[ABCDGIRSTW]]' ;;
esac

if test "$lt_cv_nm_interface" = "MS dumpbin"; then
 # Gets list of data symbols to import.
 lt_cv_sys_global_symbol_to_import="sed -n -e 's/^I .* \(.*\)$/\1/p'"
 # Adjust the below global symbol transforms to fixup imported variables.
 lt_cdecl_hook=" -e 's/^I .* \(.*\)$/extern __declspec(dllimport) char \1;/p'"
 lt_c_name_hook=" -e 's/^I .* \(.*\)$/  {\"\1\", (void *) 0},/p'"
 lt_c_name_lib_hook="\
 -e 's/^I .* \(lib.*\)$/  {\"\1\", (void *) 0},/p'\
 -e 's/^I .* \(.*\)$/  {\"lib\1\", (void *) 0},/p'"
else
 # Disable hooks by default.
 lt_cv_sys_global_symbol_to_import=
 lt_cdecl_hook=
 lt_c_name_hook=
 lt_c_name_lib_hook=
fi

# Transform an extracted symbol line into a proper C declaration.
# Some systems (esp. on ia64) link data and code symbols differently,
# so use this general approach.
lt_cv_sys_global_symbol_to_cdecl="sed -n"\
$lt_cdecl_hook\
" -e 's/^T .* \(.*\)$/extern int \1();/p'"\
" -e 's/^$symcode$symcode* .* \(.*\)$/extern char \1;/p'"

# Transform an extracted symbol line into symbol name and symbol address
lt_cv_sys_global_symbol_to_c_name_address="sed -n"\
$lt_c_name_hook\
" -e 's/^: \(.*\) .*$/  {\"\1\", (void *) 0},/p'"\
" -e 's/^$symcode$symcode* .* \(.*\)$/  {\"\1\", (void *) \&\1},/p'"

# Transform an extracted symbol line into symbol name with lib prefix and
# symbol address.
lt_cv_sys_global_symbol_to_c_name_address_lib_prefix="sed -n"\
$lt_c_name_lib_hook\
" -e 's/^: \(.*\) .*$/  {\"\1\", (void *) 0},/p'"\
" -e 's/^$symcode$symcode* .* \(lib.*\)$/  {\"\1\", (void *) \&\1},/p'"\
" -e 's/^$symcode$symcode* .* \(.*\)$/  {\"lib\1\", (void *) \&\1},/p'"

# Handle CRLF in mingw tool chain
opt_cr=
case $build_os in
mingw*)
 opt_cr=`$ECHO 'x\{0,1\}' | tr x '\015'` # option cr in regexp
 ;;
esac

# Try without a prefix underscore, then with it.
for ac_symprfx in "" "_"; do

 # Transform symcode, sympat, and symprfx into a raw symbol and a C symbol.
 symxfrm="\\1 $ac_symprfx\\2 \\2"

 # Write the raw and C identifiers.
 if test "$lt_cv_nm_interface" = "MS dumpbin"; then
   # Fake it for dumpbin and say T for any non-static function,
   # D for any global variable and I for any imported variable.
   # Also find C++ and __fastcall symbols from MSVC++,
   # which start with @ or ?.
   lt_cv_sys_global_symbol_pipe="$AWK ['"\
"     {last_section=section; section=\$ 3};"\
"     /^COFF SYMBOL TABLE/{for(i in hide) delete hide[i]};"\
"     /Section length .*#relocs.*(pick any)/{hide[last_section]=1};"\
"     /^ *Symbol name *: /{split(\$ 0,sn,\":\"); si=substr(sn[2],2)};"\
"     /^ *Type *: code/{print \"T\",si,substr(si,length(prfx))};"\
"     /^ *Type *: data/{print \"I\",si,substr(si,length(prfx))};"\
"     \$ 0!~/External *\|/{next};"\
"     / 0+ UNDEF /{next}; / UNDEF \([^|]\)*()/{next};"\
"     {if(hide[section]) next};"\
"     {f=\"D\"}; \$ 0~/\(\).*\|/{f=\"T\"};"\
"     {split(\$ 0,a,/\||\r/); split(a[2],s)};"\
"     s[1]~/^[@?]/{print f,s[1],s[1]; next};"\
"     s[1]~prfx {split(s[1],t,\"@\"); print f,t[1],substr(t[1],length(prfx))}"\
"     ' prfx=^$ac_symprfx]"
 else
   lt_cv_sys_global_symbol_pipe="sed -n -e 's/^.*[[  ]]\($symcode$symcode*\)[[  ]][[
]]*$ac_symprfx$sympat$opt_cr$/$symxfrm/p'"
 fi
 lt_cv_sys_global_symbol_pipe="$lt_cv_sys_global_symbol_pipe | sed '/ __gnu_lto/d'"
 # Check to see that the pipe works correctly.
 pipe_works=no

 rm -f conftest*
 cat > conftest.$ac_ext <<_LT_EOF
#ifdef __cplusplus
extern "C" {
#endif
char nm_test_var;
void nm_test_func(void);
void nm_test_func(void){}
#ifdef __cplusplus
}
#endif
int main(){nm_test_var='a';nm_test_func();return(0);}
_LT_EOF

 if AC_TRY_EVAL(ac_compile); then
   # Now try to grab the symbols.
   nlist=conftest.nm
   if AC_TRY_EVAL(NM conftest.$ac_objext \| "$lt_cv_sys_global_symbol_pipe" \> $nlist) && test -s "$nlist";
then
     # Try sorting and uniquifying the output.
     if sort "$nlist" | uniq > "$nlist"T; then
 mv -f "$nlist"T "$nlist"
     else
 rm -f "$nlist"T
     fi

     # Make sure that we snagged all the symbols we need.
     if $GREP ' nm_test_var$' "$nlist" >/dev/null; then
 if $GREP ' nm_test_func$' "$nlist" >/dev/null; then
   cat <<_LT_EOF > conftest.$ac_ext
/* Keep this code in sync between libtool.m4, ltmain, lt_system.h, and tests.  */
#if defined _WIN32 || defined __CYGWIN__ || defined _WIN32_WCE
/* DATA imports from DLLs on WIN32 can't be const, because runtime
  relocations are performed -- see ld's documentation on pseudo-relocs.  */
# define LT@&t@_DLSYM_CONST
#elif defined __osf__
/* This system does not cope well with relocations in const data.  */
# define LT@&t@_DLSYM_CONST
#else
# define LT@&t@_DLSYM_CONST const
#endif

#ifdef __cplusplus
extern "C" {
#endif
_LT_EOF
   # Now generate the symbol file.
   eval "$lt_cv_sys_global_symbol_to_cdecl"' < "$nlist" | $GREP -v main >> conftest.$ac_ext'

   cat <<_LT_EOF >> conftest.$ac_ext

/* The mapping between symbol names and symbols.  */
LT@&t@_DLSYM_CONST struct {
 const char *name;
 void       *address;
}
lt__PROGRAM__LTX_preloaded_symbols[[]] =
{
 { "@PROGRAM@", (void *) 0 },
_LT_EOF
   $SED "s/^$symcode$symcode* .* \(.*\)$/  {\"\1\", (void *) \&\1},/" < "$nlist" | $GREP -v main >>
conftest.$ac_ext
   cat <<\_LT_EOF >> conftest.$ac_ext
 {0, (void *) 0}
};

/* This works around a problem in FreeBSD linker */
#ifdef FREEBSD_WORKAROUND
static const void *lt_preloaded_setup() {
 return lt__PROGRAM__LTX_preloaded_symbols;
}
#endif

#ifdef __cplusplus
}
#endif
_LT_EOF
   # Now try linking the two files.
   mv -f conftest.$ac_objext conftstm.$ac_objext
   lt_globsym_save_LIBS=$LIBS
   lt_globsym_save_CFLAGS=$CFLAGS
   LIBS=conftstm.$ac_objext
   CFLAGS="$CFLAGS$_LT_TAGVAR(lt_prog_compiler_no_builtin_flag, $1)"
   if AC_TRY_EVAL(ac_link) && test -s conftest$ac_exeext; then
     pipe_works=yes
   fi
   LIBS=$lt_globsym_save_LIBS
   CFLAGS=$lt_globsym_save_CFLAGS
 else
   echo "cannot find nm_test_func in $nlist" >&AS_MESSAGE_LOG_FD
 fi
     else
 echo "cannot find nm_test_var in $nlist" >&AS_MESSAGE_LOG_FD
     fi
   else
     echo "cannot run $lt_cv_sys_global_symbol_pipe" >&AS_MESSAGE_LOG_FD
   fi
 else
   echo "$progname: failed program was:" >&AS_MESSAGE_LOG_FD
   cat conftest.$ac_ext >&5
 fi
 rm -rf conftest* conftst*

 # Do not use the global_symbol_pipe unless it works.
 if test yes = "$pipe_works"; then
   break
 else
   lt_cv_sys_global_symbol_pipe=
 fi
done
])
if test -z "$lt_cv_sys_global_symbol_pipe"; then
 lt_cv_sys_global_symbol_to_cdecl=
fi
if test -z "$lt_cv_sys_global_symbol_pipe$lt_cv_sys_global_symbol_to_cdecl"; then
 AC_MSG_RESULT(failed)
else
 AC_MSG_RESULT(ok)
fi

# Response file support.
if test "$lt_cv_nm_interface" = "MS dumpbin"; then
 nm_file_list_spec='@'
elif $NM --help 2>/dev/null | grep '[[@]]FILE' >/dev/null; then
 nm_file_list_spec='@'
fi

_LT_DECL([global_symbol_pipe], [lt_cv_sys_global_symbol_pipe], [1],
   [Take the output of nm and produce a listing of raw symbols and C names])
_LT_DECL([global_symbol_to_cdecl], [lt_cv_sys_global_symbol_to_cdecl], [1],
   [Transform the output of nm in a proper C declaration])
_LT_DECL([global_symbol_to_import], [lt_cv_sys_global_symbol_to_import], [1],
   [Transform the output of nm into a list of symbols to manually relocate])
_LT_DECL([global_symbol_to_c_name_address],
   [lt_cv_sys_global_symbol_to_c_name_address], [1],
   [Transform the output of nm in a C name address pair])
_LT_DECL([global_symbol_to_c_name_address_lib_prefix],
   [lt_cv_sys_global_symbol_to_c_name_address_lib_prefix], [1],
   [Transform the output of nm in a C name address pair when lib prefix is needed])
_LT_DECL([nm_interface], [lt_cv_nm_interface], [1],
   [The name lister interface])
_LT_DECL([], [nm_file_list_spec], [1],
   [Specify filename containing input files for $NM])
]) # _LT_CMD_GLOBAL_SYMBOLS


# _LT_COMPILER_PIC([TAGNAME])
# ---------------------------
m4_defun([_LT_COMPILER_PIC],
[m4_require([_LT_TAG_COMPILER])dnl
_LT_TAGVAR(lt_prog_compiler_wl, $1)=
_LT_TAGVAR(lt_prog_compiler_pic, $1)=
_LT_TAGVAR(lt_prog_compiler_static, $1)=

m4_if([$1], [CXX], [
 # C++ specific cases for pic, static, wl, etc.
 if test yes = "$GXX"; then
   _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
   _LT_TAGVAR(lt_prog_compiler_static, $1)='-static'

   case $host_os in
   aix*)
     # All AIX code is PIC.
     if test ia64 = "$host_cpu"; then
 # AIX 5 now supports IA64 processor
 _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
     fi
     _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
     ;;

   amigaos*)
     case $host_cpu in
     powerpc)
           # see comment about AmigaOS4 .so support
           _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
       ;;
     m68k)
           # FIXME: we need at least 68020 code to build shared libraries, but
           # adding the '-m68020' flag to GCC prevents building anything better,
           # like '-m68040'.
           _LT_TAGVAR(lt_prog_compiler_pic, $1)='-m68020 -resident32 -malways-restore-a4'
       ;;
     esac
     ;;

   beos* | irix5* | irix6* | nonstopux* | osf3* | osf4* | osf5*)
     # PIC is the default for these OSes.
     ;;
```

```
  mingw* | cygwin* | os2* | pw32* | cegcc*)
    # This hack is so that the source file can tell whether it is being
    # built for inclusion in a dll (and should export symbols for example).
    # Although the cygwin gcc ignores -fPIC, still need this for old-style
    # (--disable-auto-import) libraries
    m4_if([$1], [GCJ], [],
[_LT_TAGVAR(lt_prog_compiler_pic, $1)='-DDLL_EXPORT'])
    case $host_os in
    os2*)
_LT_TAGVAR(lt_prog_compiler_static, $1)='$wl-static'
;;
    esac
    ;;
  darwin* | rhapsody*)
    # PIC is the default on this platform
    # Common symbols not allowed in MH_DYLIB files
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fno-common'
    ;;
  *djgpp*)
    # DJGPP does not support shared libraries at all
    _LT_TAGVAR(lt_prog_compiler_pic, $1)=
    ;;
  haiku*)
    # PIC is the default for Haiku.
    # The "-static" flag exists, but is broken.
    _LT_TAGVAR(lt_prog_compiler_static, $1)=
    ;;
  interix[[3-9]]*)
    # Interix 3.x gcc -fpic/-fPIC options generate broken code.
    # Instead, we relocate shared libraries at runtime.
    ;;
  sysv4*MP*)
    if test -d /usr/nec; then
_LT_TAGVAR(lt_prog_compiler_pic, $1)=-Kconform_pic
    fi
    ;;
  hpux*)
    # PIC is the default for 64-bit PA HP-UX, but not for 32-bit
    # PA HP-UX.  On IA64 HP-UX, PIC is the default but the pic flag
    # sets the default TLS model and affects inlining.
    case $host_cpu in
    hppa*64*)
;;
    *)
_LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
;;
    esac
    ;;
```

```
   *qnx* | *nto*)
     # QNX uses GNU C++, but need to define -shared option too, otherwise
     # it will coredump.
     _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC -shared'
     ;;
   *)
     _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
     ;;
   esac
 else
   case $host_os in
     aix[[4-9]]*)
 # All AIX code is PIC.
 if test ia64 = "$host_cpu"; then
   # AIX 5 now supports IA64 processor
   _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
 else
   _LT_TAGVAR(lt_prog_compiler_static, $1)='-bnso -bI:/lib/syscalls.exp'
 fi
 ;;
     chorus*)
 case $cc_basename in
 cxch68*)
   # Green Hills C++ Compiler
   # _LT_TAGVAR(lt_prog_compiler_static, $1)="--no_auto_instantiation -u __main -u __premain -u _abort -r
$COOL_DIR/lib/libOrb.a $MVME_DIR/lib/CC/libC.a $MVME_DIR/lib/classix/libcx.s.a"
   ;;
 esac
 ;;
     mingw* | cygwin* | os2* | pw32* | cegcc*)
 # This hack is so that the source file can tell whether it is being
 # built for inclusion in a dll (and should export symbols for example).
 m4_if([$1], [GCJ], [],
   [_LT_TAGVAR(lt_prog_compiler_pic, $1)='-DDLL_EXPORT'])
 ;;
     dgux*)
 case $cc_basename in
   ec++*)
     _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
     ;;
   ghcx*)
     # Green Hills C++ Compiler
     _LT_TAGVAR(lt_prog_compiler_pic, $1)='-pic'
     ;;
   *)
     ;;
 esac
 ;;
```

```
    freebsd* | dragonfly*)
# FreeBSD uses GNU C++
;;
    hpux9* | hpux10* | hpux11*)
case $cc_basename in
  CC*)
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='$wl-a ${wl}archive'
    if test ia64 != "$host_cpu"; then
      _LT_TAGVAR(lt_prog_compiler_pic, $1)='+Z'
    fi
    ;;
  aCC*)
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='$wl-a ${wl}archive'
    case $host_cpu in
    hppa*64*|ia64*)
      # +Z the default
      ;;
    *)
      _LT_TAGVAR(lt_prog_compiler_pic, $1)='+Z'
      ;;
    esac
    ;;
  *)
    ;;
esac
;;
    interix*)
# This is c89, which is MS Visual C++ (no shared libs)
# Anyone wants to do a port?
;;
    irix5* | irix6* | nonstopux*)
case $cc_basename in
  CC*)
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-non_shared'
    # CC pic flag -KPIC is the default.
    ;;
  *)
    ;;
esac
;;
    linux* | k*bsd*-gnu | kopensolaris*-gnu | gnu*)
case $cc_basename in
  KCC*)
    # KAI C++ Compiler
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='--backend -Wl,'
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
    ;;
  ecpc* )
    # old Intel C++ for x86_64, which still supported -KPIC.
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-static'
    ;;
  icpc* )
    # Intel C++, used to be incompatible with GCC.
    # ICC 10 doesn't accept -KPIC any more.
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-static'
    ;;
  pgCC* | pgcpp*)
    # Portland Group C++ compiler
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fpic'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
    ;;
  cxx*)
    # Compaq C++
    # Make sure the PIC flag is empty.  It appears that all Alpha
    # Linux and Compaq Tru64 Unix objects are PIC.
    _LT_TAGVAR(lt_prog_compiler_pic, $1)=
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-non_shared'
    ;;
  xlc* | xlC* | bgxl[[cC]]* | mpixl[[cC]]*)
    # IBM XL 8.0, 9.0 on PPC and BlueGene
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-qpic'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-qstaticlink'
    ;;
  *)
    case `$CC -V 2>&1 | sed 5q` in
    *Sun\ C*)
      # Sun C++ 5.9
      _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
      _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
      _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Qoption ld '
      ;;
    esac
    ;;
esac
;;
    lynxos*)
;;
```

```
  m88k*)
;;
  mvs*)
case $cc_basename in
 cxx*)
  _LT_TAGVAR(lt_prog_compiler_pic, $1)='-W c,exportall'
  ;;
 *)
  ;;
esac
;;
  netbsd* | netbsdelf*-gnu)
;;
  *qnx* | *nto*)
   # QNX uses GNU C++, but need to define -shared option too, otherwise
   # it will coredump.
   _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC -shared'
   ;;
  osf3* | osf4* | osf5*)
case $cc_basename in
 KCC*)
  _LT_TAGVAR(lt_prog_compiler_wl, $1)='--backend -Wl,'
  ;;
 RCC*)
  # Rational C++ 2.4.1
  _LT_TAGVAR(lt_prog_compiler_pic, $1)='-pic'
  ;;
 cxx*)
  # Digital/Compaq C++
  _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
  # Make sure the PIC flag is empty.  It appears that all Alpha
  # Linux and Compaq Tru64 Unix objects are PIC.
  _LT_TAGVAR(lt_prog_compiler_pic, $1)=
  _LT_TAGVAR(lt_prog_compiler_static, $1)='-non_shared'
  ;;
 *)
  ;;
esac
;;
  psos*)
;;
  solaris*)
case $cc_basename in
 CC* | sunCC*)
  # Sun C++ 4.2, 5.x and Centerline C++
  _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
  _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
  _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Qoption ld '
```

```
    ;;
  gcx*)
    # Green Hills C++ Compiler
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-PIC'
    ;;
  *)
    ;;
esac
;;
    sunos4*)
case $cc_basename in
  CC*)
    # Sun C++ 4.x
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-pic'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
    ;;
  lcc*)
    # Lucid
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-pic'
    ;;
  *)
    ;;
esac
;;
    sysv5* | unixware* | sco3.2v5* | sco5v6* | OpenUNIX*)
case $cc_basename in
  CC*)
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
    ;;
esac
;;
    tandem*)
case $cc_basename in
  NCC*)
    # NonStop-UX NCC 3.20
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
    ;;
  *)
    ;;
esac
;;
    vxworks*)
;;
    *)
_LT_TAGVAR(lt_prog_compiler_can_build_shared, $1)=no
;;
```

```
    esac
  fi
],
[
  if test yes = "$GCC"; then
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-static'

    case $host_os in
      aix*)
      # All AIX code is PIC.
      if test ia64 = "$host_cpu"; then
  # AIX 5 now supports IA64 processor
  _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
      fi
      _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
      ;;

    amigaos*)
      case $host_cpu in
      powerpc)
            # see comment about AmigaOS4 .so support
            _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
        ;;
      m68k)
            # FIXME: we need at least 68020 code to build shared libraries, but
            # adding the '-m68020' flag to GCC prevents building anything better,
            # like '-m68040'.
            _LT_TAGVAR(lt_prog_compiler_pic, $1)='-m68020 -resident32 -malways-restore-a4'
        ;;
      esac
      ;;

    beos* | irix5* | irix6* | nonstopux* | osf3* | osf4* | osf5*)
      # PIC is the default for these OSes.
      ;;

    mingw* | cygwin* | pw32* | os2* | cegcc*)
      # This hack is so that the source file can tell whether it is being
      # built for inclusion in a dll (and should export symbols for example).
      # Although the cygwin gcc ignores -fPIC, still need this for old-style
      # (--disable-auto-import) libraries
      m4_if([$1], [GCJ], [],
  [_LT_TAGVAR(lt_prog_compiler_pic, $1)='-DDLL_EXPORT'])
      case $host_os in
      os2*)
  _LT_TAGVAR(lt_prog_compiler_static, $1)='$wl-static'
  ;;
```

```
    esac
    ;;

  darwin* | rhapsody*)
    # PIC is the default on this platform
    # Common symbols not allowed in MH_DYLIB files
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fno-common'
    ;;

  haiku*)
    # PIC is the default for Haiku.
    # The "-static" flag exists, but is broken.
    _LT_TAGVAR(lt_prog_compiler_static, $1)=
    ;;

  hpux*)
    # PIC is the default for 64-bit PA HP-UX, but not for 32-bit
    # PA HP-UX.  On IA64 HP-UX, PIC is the default but the pic flag
    # sets the default TLS model and affects inlining.
    case $host_cpu in
    hppa*64*)
# +Z the default
;;
    *)
_LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
;;
    esac
    ;;

  interix[[3-9]]*)
    # Interix 3.x gcc -fpic/-fPIC options generate broken code.
    # Instead, we relocate shared libraries at runtime.
    ;;

  msdosdjgpp*)
    # Just because we use GCC doesn't mean we suddenly get shared libraries
    # on systems that don't support them.
    _LT_TAGVAR(lt_prog_compiler_can_build_shared, $1)=no
    enable_shared=no
    ;;

  *nto* | *qnx*)
    # QNX uses GNU C++, but need to define -shared option too, otherwise
    # it will coredump.
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC -shared'
    ;;

  sysv4*MP*)
```

```
    if test -d /usr/nec; then
_LT_TAGVAR(lt_prog_compiler_pic, $1)=-Kconform_pic
    fi
    ;;

  *)
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
    ;;
  esac

  case $cc_basename in
  nvcc*) # Cuda Compiler Driver 2.2
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Xlinker '
    if test -n "$_LT_TAGVAR(lt_prog_compiler_pic, $1)"; then
      _LT_TAGVAR(lt_prog_compiler_pic, $1)="-Xcompiler $_LT_TAGVAR(lt_prog_compiler_pic, $1)"
    fi
    ;;
  esac
else
  # PORTME Check for flag to pass linker flags through the system compiler.
  case $host_os in
  aix*)
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    if test ia64 = "$host_cpu"; then
# AIX 5 now supports IA64 processor
_LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
    else
_LT_TAGVAR(lt_prog_compiler_static, $1)='-bnso -bI:/lib/syscalls.exp'
    fi
    ;;

  darwin* | rhapsody*)
    # PIC is the default on this platform
    # Common symbols not allowed in MH_DYLIB files
    _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fno-common'
    case $cc_basename in
    nagfor*)
      # NAG Fortran compiler
      _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,-Wl,,'
      _LT_TAGVAR(lt_prog_compiler_pic, $1)='-PIC'
      _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
      ;;
    esac
    ;;

  mingw* | cygwin* | pw32* | os2* | cegcc*)
    # This hack is so that the source file can tell whether it is being
    # built for inclusion in a dll (and should export symbols for example).
    m4_if([$1], [GCJ], [],
[_LT_TAGVAR(lt_prog_compiler_pic, $1)='-DDLL_EXPORT'])
    case $host_os in
    os2*)
_LT_TAGVAR(lt_prog_compiler_static, $1)='$wl-static'
;;
    esac
    ;;

  hpux9* | hpux10* | hpux11*)
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    # PIC is the default for IA64 HP-UX and 64-bit HP-UX, but
    # not for PA HP-UX.
    case $host_cpu in
    hppa*64*|ia64*)
# +Z the default
;;
    *)
_LT_TAGVAR(lt_prog_compiler_pic, $1)='+Z'
;;
    esac
    # Is there a better lt_prog_compiler_static that works with the bundled CC?
    _LT_TAGVAR(lt_prog_compiler_static, $1)='$wl-a ${wl}archive'
    ;;

  irix5* | irix6* | nonstopux*)
    _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
    # PIC (with -KPIC) is the default.
    _LT_TAGVAR(lt_prog_compiler_static, $1)='-non_shared'
    ;;

  linux* | k*bsd*-gnu | kopensolaris*-gnu | gnu*)
    case $cc_basename in
    # old Intel for x86_64, which still supported -KPIC.
    ecc*)
_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
_LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
_LT_TAGVAR(lt_prog_compiler_static, $1)='-static'
      ;;
    # icc used to be incompatible with GCC.
    # ICC 10 doesn't accept -KPIC any more.
    icc* | ifort*)
_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
_LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
_LT_TAGVAR(lt_prog_compiler_static, $1)='-static'
      ;;
    # Lahey Fortran 8.1.
    lf95*)
```

```
_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
_LT_TAGVAR(lt_prog_compiler_pic, $1)='--shared'
_LT_TAGVAR(lt_prog_compiler_static, $1)='--static'
;;
  nagfor*)
# NAG Fortran compiler
_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,-Wl,,'
_LT_TAGVAR(lt_prog_compiler_pic, $1)='-PIC'
_LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
;;
  tcc*)
# Fabrice Bellard et al's Tiny C Compiler
_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
_LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
_LT_TAGVAR(lt_prog_compiler_static, $1)='-static'
;;
  pgcc* | pgf77* | pgf90* | pgf95* | pgfortran*)
   # Portland Group compilers (*not* the Pentium gcc compiler,
# which looks to be a dead project)
_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
_LT_TAGVAR(lt_prog_compiler_pic, $1)='-fpic'
_LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
   ;;
  ccc*)
   _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
   # All Alpha code is PIC.
   _LT_TAGVAR(lt_prog_compiler_static, $1)='-non_shared'
   ;;
  xl* | bgxl* | bgf* | mpixl*)
# IBM XL C 8.0/Fortran 10.1, 11.1 on PPC and BlueGene
_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
_LT_TAGVAR(lt_prog_compiler_pic, $1)='-qpic'
_LT_TAGVAR(lt_prog_compiler_static, $1)='-qstaticlink'
;;
  *)
case `$CC -V 2>&1 | sed 5q` in
*Sun\ Ceres\ Fortran* | *Sun*Fortran*\ [[1-7]].* | *Sun*Fortran*\ 8.[[0-3]]*)
 # Sun Fortran 8.3 passes all unrecognized flags to the linker
 _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
 _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
 _LT_TAGVAR(lt_prog_compiler_wl, $1)=''
 ;;
*Sun\ F* | *Sun*Fortran*)
 _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
 _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
 _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Qoption ld '
 ;;
*Sun\ C*)
```

```
 # Sun C 5.9
 _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
 _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
 _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
 ;;
    *Intel*\ [[CF]]*Compiler*)
 _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
 _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC'
 _LT_TAGVAR(lt_prog_compiler_static, $1)='-static'
 ;;
*Portland\ Group*)
 _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
 _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fpic'
 _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
 ;;
esac
;;
   esac
   ;;

 newsos6)
   _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
   _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
   ;;

 *nto* | *qnx*)
   # QNX uses GNU C++, but need to define -shared option too, otherwise
   # it will coredump.
   _LT_TAGVAR(lt_prog_compiler_pic, $1)='-fPIC -shared'
   ;;

 osf3* | osf4* | osf5*)
   _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
   # All OSF/1 code is PIC.
   _LT_TAGVAR(lt_prog_compiler_static, $1)='-non_shared'
   ;;

 rdos*)
   _LT_TAGVAR(lt_prog_compiler_static, $1)='-non_shared'
   ;;

 solaris*)
   _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
   _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
   case $cc_basename in
   f77* | f90* | f95* | sunf77* | sunf90* | sunf95*)
_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Qoption ld ';;
   *)
	_LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,';;
	  esac
	  ;;

	sunos4*)
	  _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Qoption ld '
	  _LT_TAGVAR(lt_prog_compiler_pic, $1)='-PIC'
	  _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
	  ;;

	sysv4 | sysv4.2uw2* | sysv4.3*)
	  _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
	  _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
	  _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
	  ;;

	sysv4*MP*)
	  if test -d /usr/nec; then
	_LT_TAGVAR(lt_prog_compiler_pic, $1)='-Kconform_pic'
	_LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
	  fi
	  ;;

	sysv5* | unixware* | sco3.2v5* | sco5v6* | OpenUNIX*)
	  _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
	  _LT_TAGVAR(lt_prog_compiler_pic, $1)='-KPIC'
	  _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
	  ;;

	unicos*)
	  _LT_TAGVAR(lt_prog_compiler_wl, $1)='-Wl,'
	  _LT_TAGVAR(lt_prog_compiler_can_build_shared, $1)=no
	  ;;

	uts4*)
	  _LT_TAGVAR(lt_prog_compiler_pic, $1)='-pic'
	  _LT_TAGVAR(lt_prog_compiler_static, $1)='-Bstatic'
	  ;;

	*)
	  _LT_TAGVAR(lt_prog_compiler_can_build_shared, $1)=no
	  ;;
	esac
 fi
])
case $host_os in
 # For platforms that do not support PIC, -DPIC is meaningless:
 *djgpp*)
```

```
   _LT_TAGVAR(lt_prog_compiler_pic, $1)=
   ;;
 *)
   _LT_TAGVAR(lt_prog_compiler_pic, $1)="$_LT_TAGVAR(lt_prog_compiler_pic, $1)@&t@m4_if([$1],[],[ -DPIC],[m4_if([$1],[CXX],[ -DPIC],[])])"
   ;;
esac

AC_CACHE_CHECK([for $compiler option to produce PIC],
 [_LT_TAGVAR(lt_cv_prog_compiler_pic, $1)],
 [_LT_TAGVAR(lt_cv_prog_compiler_pic, $1)=$_LT_TAGVAR(lt_prog_compiler_pic, $1)])
_LT_TAGVAR(lt_prog_compiler_pic, $1)=$_LT_TAGVAR(lt_cv_prog_compiler_pic, $1)

#
# Check to make sure the PIC flag actually works.
#
if test -n "$_LT_TAGVAR(lt_prog_compiler_pic, $1)"; then
 _LT_COMPILER_OPTION([if $compiler PIC flag $_LT_TAGVAR(lt_prog_compiler_pic, $1) works],
   [_LT_TAGVAR(lt_cv_prog_compiler_pic_works, $1)],
   [$_LT_TAGVAR(lt_prog_compiler_pic, $1)@&t@m4_if([$1],[],[ -DPIC],[m4_if([$1],[CXX],[ -DPIC],[])])], [],
   [case $_LT_TAGVAR(lt_prog_compiler_pic, $1) in
    "" | " "*) ;;
    *) _LT_TAGVAR(lt_prog_compiler_pic, $1)=" $_LT_TAGVAR(lt_prog_compiler_pic, $1)" ;;
    esac],
   [_LT_TAGVAR(lt_prog_compiler_pic, $1)=
    _LT_TAGVAR(lt_prog_compiler_can_build_shared, $1)=no])
fi
_LT_TAGDECL([pic_flag], [lt_prog_compiler_pic], [1],
 [Additional compiler flags for building library objects])

_LT_TAGDECL([wl], [lt_prog_compiler_wl], [1],
 [How to pass a linker flag through the compiler])
#
# Check to make sure the static flag actually works.
#
wl=$_LT_TAGVAR(lt_prog_compiler_wl, $1) eval lt_tmp_static_flag=\"$_LT_TAGVAR(lt_prog_compiler_static, $1)\"
_LT_LINKER_OPTION([if $compiler static flag $lt_tmp_static_flag works],
 _LT_TAGVAR(lt_cv_prog_compiler_static_works, $1),
 $lt_tmp_static_flag,
 [],
 [_LT_TAGVAR(lt_prog_compiler_static, $1)=])
_LT_TAGDECL([link_static_flag], [lt_prog_compiler_static], [1],
 [Compiler flag to prevent dynamic linking])
])# _LT_COMPILER_PIC


# _LT_LINKER_SHLIBS([TAGNAME])
# ----------------------------
# See if the linker supports building shared libraries.
m4_defun([_LT_LINKER_SHLIBS],
[AC_REQUIRE([LT_PATH_LD])dnl
AC_REQUIRE([LT_PATH_NM])dnl
m4_require([_LT_PATH_MANIFEST_TOOL])dnl
m4_require([_LT_FILEUTILS_DEFAULTS])dnl
m4_require([_LT_DECL_EGREP])dnl
m4_require([_LT_DECL_SED])dnl
m4_require([_LT_CMD_GLOBAL_SYMBOLS])dnl
m4_require([_LT_TAG_COMPILER])dnl
AC_MSG_CHECKING([whether the $compiler linker ($LD) supports shared libraries])
m4_if([$1], [CXX], [
 _LT_TAGVAR(export_symbols_cmds, $1)='$NM $libobjs $convenience | $global_symbol_pipe | $SED '\''s/.* //'\''
| sort | uniq > $export_symbols'
 _LT_TAGVAR(exclude_expsyms, $1)=['_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*']
 case $host_os in
 aix[[4-9]]*)
   # If we're using GNU nm, then we don't want the "-C" option.
   # -C means demangle to GNU nm, but means don't demangle to AIX nm.
   # Without the "-l" option, or with the "-B" option, AIX nm treats
   # weak defined symbols like other global defined symbols, whereas
   # GNU nm marks them as "W".
   # While the 'weak' keyword is ignored in the Export File, we need
   # it in the Import File for the 'aix-soname' feature, so we have
   # to replace the "-B" option with "-P" for AIX nm.
   if $NM -V 2>&1 | $GREP 'GNU' > /dev/null; then
     _LT_TAGVAR(export_symbols_cmds, $1)='$NM -Bpg $libobjs $convenience | awk '\''{ if (((\$ 2 == "T") || (\$
2 == "D") || (\$ 2 == "B") || (\$ 2 == "W")) && ([substr](\$ 3,1,1) != ".")) { if (\$ 2 == "W") { print \$ 3 " weak" }
else { print \$ 3 } } }'\'' | sort -u > $export_symbols'
   else
     _LT_TAGVAR(export_symbols_cmds, $1)='`func_echo_all $NM | $SED -e '\''s/B\([[^B]]*\)$/P\1/'\''` -PCpgl
$libobjs $convenience | awk '\''{ if (((\$ 2 == "T") || (\$ 2 == "D") || (\$ 2 == "B") || (\$ 2 == "W") || (\$ 2 == "V") ||
(\$ 2 == "Z")) && ([substr](\$ 1,1,1) != ".")) { if ((\$ 2 == "W") || (\$ 2 == "V") || (\$ 2 == "Z")) { print \$ 1 " weak"
} else { print \$ 1 } } }'\'' | sort -u > $export_symbols'
   fi
   ;;
 pw32*)
   _LT_TAGVAR(export_symbols_cmds, $1)=$ltdll_cmds
   ;;
 cygwin* | mingw* | cegcc*)
   case $cc_basename in
   cl*)
     _LT_TAGVAR(exclude_expsyms, $1)='_NULL_IMPORT_DESCRIPTOR|_IMPORT_DESCRIPTOR_.*'
     ;;
   *)
     _LT_TAGVAR(export_symbols_cmds, $1)='$NM $libobjs $convenience | $global_symbol_pipe | $SED -e
'\''/^[[BCDGRS]][[ ]]/s/.*[[ ]]\([[^ ]]*\)/\1 DATA/;s/^.*[[ ]]__nm__\([[^ ]]*\)[[ ]][[^ ]]*/\1 DATA/;/^I[[
]]/d;/^[[AITW]][[ ]]/s/.* //'\'' | sort | uniq > $export_symbols'
     _LT_TAGVAR(exclude_expsyms,
$1)=['[_]+GLOBAL_OFFSET_TABLE_|[_]+GLOBAL__[FID]_.*|[_]+head_[A-Za-z0-9_]+_dll|[A-Za-z0-
9_]+_dll_iname']
     ;;
   esac
   ;;
 linux* | k*bsd*-gnu | gnu*)
   _LT_TAGVAR(link_all_deplibs, $1)=no
   ;;
 *)
   _LT_TAGVAR(export_symbols_cmds, $1)='$NM $libobjs $convenience | $global_symbol_pipe | $SED '\''s/.*
//'\'' | sort | uniq > $export_symbols'
   ;;
 esac
], [
 runpath_var=
 _LT_TAGVAR(allow_undefined_flag, $1)=
 _LT_TAGVAR(always_export_symbols, $1)=no
 _LT_TAGVAR(archive_cmds, $1)=
 _LT_TAGVAR(archive_expsym_cmds, $1)=
 _LT_TAGVAR(compiler_needs_object, $1)=no
 _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=no
 _LT_TAGVAR(export_dynamic_flag_spec, $1)=
 _LT_TAGVAR(export_symbols_cmds, $1)='$NM $libobjs $convenience | $global_symbol_pipe | $SED '\''s/.* //'\''
| sort | uniq > $export_symbols'
 _LT_TAGVAR(hardcode_automatic, $1)=no
 _LT_TAGVAR(hardcode_direct, $1)=no
 _LT_TAGVAR(hardcode_direct_absolute, $1)=no
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)=
 _LT_TAGVAR(hardcode_libdir_separator, $1)=
 _LT_TAGVAR(hardcode_minus_L, $1)=no
 _LT_TAGVAR(hardcode_shlibpath_var, $1)=unsupported
 _LT_TAGVAR(inherit_rpath, $1)=no
 _LT_TAGVAR(link_all_deplibs, $1)=unknown
 _LT_TAGVAR(module_cmds, $1)=
 _LT_TAGVAR(module_expsym_cmds, $1)=
 _LT_TAGVAR(old_archive_from_new_cmds, $1)=
 _LT_TAGVAR(old_archive_from_expsyms_cmds, $1)=
 _LT_TAGVAR(thread_safe_flag_spec, $1)=
 _LT_TAGVAR(whole_archive_flag_spec, $1)=
 # include_expsyms should be a list of space-separated symbols to be *always*
 # included in the symbol list
 _LT_TAGVAR(include_expsyms, $1)=
 # exclude_expsyms can be an extended regexp of symbols to exclude
 # it will be wrapped by ' (' and ')$', so one must not match beginning or
 # end of line.  Example: 'a|bc|.*d.*' will exclude the symbols 'a' and 'bc',
 # as well as any symbol that contains 'd'.
 _LT_TAGVAR(exclude_expsyms, $1)=['_GLOBAL_OFFSET_TABLE_|_GLOBAL__F[ID]_.*']
 # Although _GLOBAL_OFFSET_TABLE_ is a valid symbol C name, most a.out
 # platforms (ab)use it in PIC code, but their linkers get confused if
 # the symbol is explicitly referenced.  Since portable code cannot
 # rely on this symbol name, it's probably fine to never include it in
 # preloaded symbol tables.
 # Exclude shared library initialization/finalization symbols.
dnl Note also adjust exclude_expsyms for C++ above.
 extract_expsyms_cmds=

 case $host_os in
 cygwin* | mingw* | pw32* | cegcc*)
   # FIXME: the MSVC++ port hasn't been tested in a loooong time
   # When not using gcc, we currently assume that we are using
   # Microsoft Visual C++.
   if test yes != "$GCC"; then
     with_gnu_ld=no
   fi
   ;;
 interix*)
   # we just hope/assume this is gcc and not c89 (= MSVC++)
   with_gnu_ld=yes
   ;;
 openbsd* | bitrig*)
   with_gnu_ld=no
   ;;
 linux* | k*bsd*-gnu | gnu*)
   _LT_TAGVAR(link_all_deplibs, $1)=no
   ;;
 esac

 _LT_TAGVAR(ld_shlibs, $1)=yes

 # On some targets, GNU ld is compatible enough with the native linker
 # that we're better off using the native interface for both.
 lt_use_gnu_ld_interface=no
 if test yes = "$with_gnu_ld"; then
   case $host_os in
     aix*)
 # The AIX port of GNU ld has always aspired to compatibility
 # with the native linker.  However, as the warning in the GNU ld
 # block says, versions before 2.19.5* couldn't really create working
 # shared libraries, regardless of the interface used.
 case `$LD -v 2>&1` in
   *\ \(GNU\ Binutils\)\ 2.19.5*) ;;
   *\ \(GNU\ Binutils\)\ 2.[[2-9]]*) ;;
   *\ \(GNU\ Binutils\)\ [[3-9]]*) ;;
   *)
```

```
    lt_use_gnu_ld_interface=yes
    ;;
  esac
  ;;
    *)
  lt_use_gnu_ld_interface=yes
  ;;
    esac
  fi

  if test yes = "$lt_use_gnu_ld_interface"; then
    # If archive_cmds runs LD, not CC, wlarc should be empty
    wlarc='$wl'

    # Set some defaults for GNU ld with shared library support. These
    # are reset later if shared libraries are not supported. Putting them
    # here allows them to be overridden if necessary.
    runpath_var=LD_RUN_PATH
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
    _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl--export-dynamic'
    # ancient GNU ld didn't support --whole-archive et. al.
    if $LD --help 2>&1 | $GREP 'no-whole-archive' > /dev/null; then
      _LT_TAGVAR(whole_archive_flag_spec, $1)=$wlarc'--whole-archive$convenience '$wlarc'--no-whole-archive'
    else
      _LT_TAGVAR(whole_archive_flag_spec, $1)=
    fi
    supports_anon_versioning=no
    case `$LD -v | $SED -e 's/([^)]\+)\s\+//' 2>&1` in
      *GNU\ gold*) supports_anon_versioning=yes ;;
      *\ [[01]].* | *\ 2.[[0-9]].* | *\ 2.10.*) ;; # catch versions < 2.11
      *\ 2.11.93.0.2\ *) supports_anon_versioning=yes ;; # RH7.3 ...
      *\ 2.11.92.0.12\ *) supports_anon_versioning=yes ;; # Mandrake 8.2 ...
      *\ 2.11.*) ;; # other 2.11 versions
      *) supports_anon_versioning=yes ;;
    esac

    # See if GNU ld supports shared libraries.
    case $host_os in
    aix[[3-9]]*)
      # On AIX/PPC, the GNU linker is very broken
      if test ia64 != "$host_cpu"; then
  _LT_TAGVAR(ld_shlibs, $1)=no
  cat <<_LT_EOF 1>&2

*** Warning: the GNU linker, at least up to release 2.19, is reported
*** to be unable to reliably create shared libraries on AIX.
*** Therefore, libtool is disabling shared libraries support.  If you
*** really care for shared libraries, you may want to install binutils
*** 2.20 or above, or modify your PATH so that a non-GNU linker is found.
*** You will then need to restart the configuration process.

_LT_EOF
      fi
      ;;

    amigaos*)
      case $host_cpu in
      powerpc)
          # see comment about AmigaOS4 .so support
          _LT_TAGVAR(archive_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname -o $lib'
          _LT_TAGVAR(archive_expsym_cmds, $1)=''
        ;;
      m68k)
          _LT_TAGVAR(archive_cmds, $1)='$RM $output_objdir/a2ixlibrary.data~$ECHO "#define NAME
$libname" > $output_objdir/a2ixlibrary.data~$ECHO "#define LIBRARY_ID 1" >>
$output_objdir/a2ixlibrary.data~$ECHO "#define VERSION $major" >> $output_objdir/a2ixlibrary.data~$ECHO
"#define REVISION $revision" >> $output_objdir/a2ixlibrary.data~$AR $AR_FLAGS $lib $libobjs~$RANLIB
$lib~(cd $output_objdir && a2ixlibrary -32)'
          _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
          _LT_TAGVAR(hardcode_minus_L, $1)=yes
        ;;
      esac
      ;;

    beos*)
      if $LD --help 2>&1 | $GREP ': supported targets:.* elf' > /dev/null; then
  _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
  # Joseph Beckenbach <jrb3@best.com> says some releases of gcc
  # support --undefined.  This deserves some investigation.  FIXME
  _LT_TAGVAR(archive_cmds, $1)='$CC -nostart $libobjs $deplibs $compiler_flags $wl-soname $wl$soname -o
$lib'
      else
  _LT_TAGVAR(ld_shlibs, $1)=no
      fi
      ;;

    cygwin* | mingw* | pw32* | cegcc*)
      # _LT_TAGVAR(hardcode_libdir_flag_spec, $1) is actually meaningless,
      # as there is no search path for DLLs.
      _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
      _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl--export-all-symbols'
      _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
      _LT_TAGVAR(always_export_symbols, $1)=no
      _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=yes
      _LT_TAGVAR(export_symbols_cmds, $1)='$NM $libobjs $convenience | $global_symbol_pipe | $SED -e
```

```
'\"/^[[BCDGRS]][[ ]]/s/.*[[ ]]\([[^ ]]*\)/\1 DATA/;s/^.*[[ ]]__nm__\([[^ ]]*\)[[ ]][[^ ]]*/\1 DATA/;/^I[[
]]/d;/^[[AITW]][[ ]]/s/.* //'\" | sort | uniq > $export_symbols'
     _LT_TAGVAR(exclude_expsyms,
$1)=['[_]+GLOBAL_OFFSET_TABLE_|[_]+GLOBAL__[FID]_.*|[_]+head_[A-Za-z0-9_]+_dll|[A-Za-z0-
9_]+_dll_iname']

     if $LD --help 2>&1 | $GREP 'auto-import' > /dev/null; then
       _LT_TAGVAR(archive_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags -o $output_objdir/$soname
$wl--enable-auto-image-base -Xlinker --out-implib -Xlinker $lib'
 # If the export-symbols file already is a .def file, use it as
 # is; otherwise, prepend EXPORTS...
 _LT_TAGVAR(archive_expsym_cmds, $1)='if _LT_DLL_DEF_P([$export_symbols]); then
         cp $export_symbols $output_objdir/$soname.def;
       else
         echo EXPORTS > $output_objdir/$soname.def;
         cat $export_symbols >> $output_objdir/$soname.def;
       fi~
       $CC -shared $output_objdir/$soname.def $libobjs $deplibs $compiler_flags -o $output_objdir/$soname $wl--
enable-auto-image-base -Xlinker --out-implib -Xlinker $lib'
     else
 _LT_TAGVAR(ld_shlibs, $1)=no
     fi
     ;;

   haiku*)
     _LT_TAGVAR(archive_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags $wl-soname $wl$soname -o
$lib'
     _LT_TAGVAR(link_all_deplibs, $1)=yes
     ;;

   os2*)
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
     _LT_TAGVAR(hardcode_minus_L, $1)=yes
     _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
     shrext_cmds=.dll
     _LT_TAGVAR(archive_cmds, $1)='$ECHO "LIBRARY ${soname%$shared_ext} INITINSTANCE
TERMINSTANCE" > $output_objdir/$libname.def~
 $ECHO "DESCRIPTION \"$libname\"" >> $output_objdir/$libname.def~
 $ECHO "DATA MULTIPLE NONSHARED" >> $output_objdir/$libname.def~
 $ECHO EXPORTS >> $output_objdir/$libname.def~
 emxexp $libobjs | $SED /"_DLL_InitTerm"/d >> $output_objdir/$libname.def~
 $CC -Zdll -Zcrtdll -o $output_objdir/$soname $libobjs $deplibs $compiler_flags $output_objdir/$libname.def~
 emximp -o $lib $output_objdir/$libname.def'
     _LT_TAGVAR(archive_expsym_cmds, $1)='$ECHO "LIBRARY ${soname%$shared_ext} INITINSTANCE
TERMINSTANCE" > $output_objdir/$libname.def~
 $ECHO "DESCRIPTION \"$libname\"" >> $output_objdir/$libname.def~
 $ECHO "DATA MULTIPLE NONSHARED" >> $output_objdir/$libname.def~
 $ECHO EXPORTS >> $output_objdir/$libname.def~
 prefix_cmds="$SED"~
 if test EXPORTS = "`$SED 1q $export_symbols`"; then
   prefix_cmds="$prefix_cmds -e 1d";
 fi~
 prefix_cmds="$prefix_cmds -e \"s/^\(.*\)$/_\1/g\""~
 cat $export_symbols | $prefix_cmds >> $output_objdir/$libname.def~
 $CC -Zdll -Zcrtdll -o $output_objdir/$soname $libobjs $deplibs $compiler_flags $output_objdir/$libname.def~
 emximp -o $lib $output_objdir/$libname.def'
     _LT_TAGVAR(old_archive_From_new_cmds, $1)='emximp -o $output_objdir/${libname}_dll.a
$output_objdir/$libname.def'
     _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=yes
     ;;

   interix[[3-9]]*)
     _LT_TAGVAR(hardcode_direct, $1)=no
     _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath,$libdir'
     _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'
     # Hack: On Interix 3.x, we cannot compile PIC because of a broken gcc.
     # Instead, shared libraries are loaded at an image base (0x10000000 by
     # default) and relocated if they conflict, which is a slow very memory
     # consuming and fragmenting process.  To avoid this, we pick a random,
     # 256 KiB-aligned image base between 0x50000000 and 0x6FFC0000 at link
     # time.  Moving up from 0x10000000 also allows more sbrk(2) space.
     _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-h,$soname
$wl--image-base,`expr ${RANDOM-$$} % 4096 / 2 \* 262144 + 1342177280` -o $lib'
     _LT_TAGVAR(archive_expsym_cmds, $1)='sed "s|^|_|" $export_symbols
>$output_objdir/$soname.expsym~$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-h,$soname $wl--
retain-symbols-file,$output_objdir/$soname.expsym $wl--image-base,`expr ${RANDOM-$$} % 4096 / 2 \* 262144
+ 1342177280` -o $lib'
     ;;

   gnu* | linux* | tpf* | k*bsd*-gnu | kopensolaris*-gnu)
     tmp_diet=no
     if test linux-dietlibc = "$host_os"; then
 case $cc_basename in
   diet\ *) tmp_diet=yes;; # linux-dietlibc with static linking (!diet-dyn)
 esac
     fi
     if $LD --help 2>&1 | $EGREP ': supported targets:.* elf' > /dev/null \
  && test no = "$tmp_diet"
     then
 tmp_addflag=' $pic_flag'
 tmp_sharedflag='-shared'
 case $cc_basename,$host_cpu in
       pgcc*)    # Portland Group C compiler
   _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive`for conv in $convenience\"\"; do test  -n
\"$conv\" && new_convenience=\"$new_convenience,$conv\"; done; func_echo_all \"$new_convenience\"` $wl--
```

```
no-whole-archive'
  tmp_addflag=' $pic_flag'
  ;;
 pgf77* | pgf90* | pgf95* | pgfortran*)
    # Portland Group f77 and f90 compilers
  _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive`for conv in $convenience\"\"; do test  -n
\"$conv\" && new_convenience=\"$new_convenience,$conv\"; done; func_echo_all \"$new_convenience\"` $wl--
no-whole-archive'
  tmp_addflag=' $pic_flag -Mnomain' ;;
 ecc*,ia64* | icc*,ia64*) # Intel C compiler on ia64
  tmp_addflag=' -i_dynamic' ;;
 efc*,ia64* | ifort*,ia64*) # Intel Fortran compiler on ia64
  tmp_addflag=' -i_dynamic -nofor_main' ;;
 ifc* | ifort*)   # Intel Fortran compiler
  tmp_addflag=' -nofor_main' ;;
 lf95*)    # Lahey Fortran 8.1
  _LT_TAGVAR(whole_archive_flag_spec, $1)=
  tmp_sharedflag='--shared' ;;
      nagfor*)                        # NAGFOR 5.3
        tmp_sharedflag='-Wl,-shared' ;;
 xl[[cC]]* | bgxl[[cC]]* | mpixl[[cC]]*) # IBM XL C 8.0 on PPC (deal with xlf below)
  tmp_sharedflag='-qmkshrobj'
  tmp_addflag= ;;
 nvcc*) # Cuda Compiler Driver 2.2
  _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive`for conv in $convenience\"\"; do test  -n
\"$conv\" && new_convenience=\"$new_convenience,$conv\"; done; func_echo_all \"$new_convenience\"` $wl--
no-whole-archive'
  _LT_TAGVAR(compiler_needs_object, $1)=yes
  ;;
 esac
 case `$CC -V 2>&1 | sed 5q` in
 *Sun\ C*)   # Sun C 5.9
  _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive`new_convenience=; for conv in
$convenience\"\"; do test -z \"$conv\" || new_convenience=\"$new_convenience,$conv\"; done; func_echo_all
\"$new_convenience\"` $wl--no-whole-archive'
  _LT_TAGVAR(compiler_needs_object, $1)=yes
  tmp_sharedflag='-G' ;;
 *Sun\ F*)   # Sun Fortran 8.3
  tmp_sharedflag='-G' ;;
 esac
 _LT_TAGVAR(archive_cmds, $1)='$CC '"$tmp_sharedflag""$tmp_addflag"' $libobjs $deplibs $compiler_flags
$wl-soname $wl$soname -o $lib'

      if test yes = "$supports_anon_versioning"; then
        _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $output_objdir/$libname.ver~
          cat $export_symbols | sed -e "s/\(.*\)/\1;/" >> $output_objdir/$libname.ver~
          echo "local: *; };" >> $output_objdir/$libname.ver~
          $CC '"$tmp_sharedflag""$tmp_addflag"' $libobjs $deplibs $compiler_flags $wl-soname $wl$soname $wl-
```

```
version-script $wl$output_objdir/$libname.ver -o $lib'
      fi

 case $cc_basename in
 tcc*)
   _LT_TAGVAR(export_dynamic_flag_spec, $1)='-rdynamic'
   ;;
 xlf* | bgf* | bgxlf* | mpixlf*)
   # IBM XL Fortran 10.1 on PPC cannot create shared libs itself
   _LT_TAGVAR(whole_archive_flag_spec, $1)='--whole-archive$convenience --no-whole-archive'
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
   _LT_TAGVAR(archive_cmds, $1)='$LD -shared $libobjs $deplibs $linker_flags -soname $soname -o $lib'
   if test yes = "$supports_anon_versioning"; then
     _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $output_objdir/$libname.ver~
          cat $export_symbols | sed -e "s/\(.*\)/\1;/" >> $output_objdir/$libname.ver~
          echo "local: *; };" >> $output_objdir/$libname.ver~
          $LD -shared $libobjs $deplibs $linker_flags -soname $soname -version-script $output_objdir/$libname.ver -
o $lib'
   fi
   ;;
 esac
     else
       _LT_TAGVAR(ld_shlibs, $1)=no
     fi
     ;;

   netbsd* | netbsdelf*-gnu)
     if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
 _LT_TAGVAR(archive_cmds, $1)='$LD -Bshareable $libobjs $deplibs $linker_flags -o $lib'
 wlarc=
     else
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname -o $lib'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'
     fi
     ;;

   solaris*)
     if $LD -v 2>&1 | $GREP 'BFD 2\.8' > /dev/null; then
 _LT_TAGVAR(ld_shlibs, $1)=no
 cat <<_LT_EOF 1>&2

*** Warning: The releases 2.8.* of the GNU linker cannot reliably
*** create shared libraries on Solaris systems.  Therefore, libtool
*** is disabling shared libraries support.  We urge you to upgrade GNU
*** binutils to release 2.9.1 or newer.  Another option is to modify
*** your PATH or compiler configuration so that the native linker is
*** used, and then restart.

_LT_EOF
     elif $LD --help 2>&1 | $GREP ': supported targets:.* elf' > /dev/null; then
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname -o $lib'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'
     else
 _LT_TAGVAR(ld_shlibs, $1)=no
     fi
     ;;

   sysv5* | sco3.2v5* | sco5v6* | unixware* | OpenUNIX*)
     case `$LD -v 2>&1` in
       *\ [[01]].* | *\ 2.[[0-9]].* | *\ 2.1[[0-5]].*)
 _LT_TAGVAR(ld_shlibs, $1)=no
 cat <<_LT_EOF 1>&2

*** Warning: Releases of the GNU linker prior to 2.16.91.0.3 cannot
*** reliably create shared libraries on SCO systems.  Therefore, libtool
*** is disabling shared libraries support.  We urge you to upgrade GNU
*** binutils to release 2.16.91.0.3 or newer.  Another option is to modify
*** your PATH or compiler configuration so that the native linker is
*** used, and then restart.

_LT_EOF
 ;;
 *)
   # For security reasons, it is highly recommended that you always
   # use absolute paths for naming shared libraries, and exclude the
   # DT_RUNPATH tag from executables and libraries.  But doing so
   # requires that you compile everything twice, which is a pain.
   if $LD --help 2>&1 | $GREP ': supported targets:.* elf' > /dev/null; then
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
     _LT_TAGVAR(archive_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags $wl-soname $wl$soname -o
$lib'
     _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'
   else
     _LT_TAGVAR(ld_shlibs, $1)=no
   fi
 ;;
     esac
     ;;

   sunos4*)
     _LT_TAGVAR(archive_cmds, $1)='$LD -assert pure-text -Bshareable -o $lib $libobjs $deplibs $linker_flags'
     wlarc=
     _LT_TAGVAR(hardcode_direct, $1)=yes
     _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
     ;;

   *)
     if $LD --help 2>&1 | $GREP ': supported targets:.* elf' > /dev/null; then
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname -o $lib'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'
     else
 _LT_TAGVAR(ld_shlibs, $1)=no
     fi
     ;;
   esac

   if test no = "$_LT_TAGVAR(ld_shlibs, $1)"; then
     runpath_var=
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)=
     _LT_TAGVAR(export_dynamic_flag_spec, $1)=
     _LT_TAGVAR(whole_archive_flag_spec, $1)=
   fi
 else
   # PORTME fill in a description of your system's linker (not GNU ld)
   case $host_os in
   aix3*)
     _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
     _LT_TAGVAR(always_export_symbols, $1)=yes
     _LT_TAGVAR(archive_expsym_cmds, $1)='$LD -o $output_objdir/$soname $libobjs $deplibs $linker_flags -
bE:$export_symbols -T512 -H512 -bM:SRE~$AR $AR_FLAGS $lib $output_objdir/$soname'
     # Note: this linker hardcodes the directories in LIBPATH if there
     # are no directories specified by -L.
     _LT_TAGVAR(hardcode_minus_L, $1)=yes
     if test yes = "$GCC" && test -z "$lt_prog_compiler_static"; then
 # Neither direct hardcoding nor static linking is supported with a
 # broken collect2.
 _LT_TAGVAR(hardcode_direct, $1)=unsupported
     fi
     ;;

   aix[[4-9]]*)
     if test ia64 = "$host_cpu"; then
 # On IA64, the linker does run time linking by default, so we don't
 # have to do anything special.
 aix_use_runtimelinking=no
 exp_sym_flag='-Bexport'
 no_entry_flag=
     else
 # If we're using GNU nm, then we don't want the "-C" option.
 # -C means demangle to GNU nm, but means don't demangle to AIX nm.
 # Without the "-l" option, or with the "-B" option, AIX nm treats
 # weak defined symbols like other global defined symbols, whereas
 # GNU nm marks them as "W".
 # While the 'weak' keyword is ignored in the Export File, we need
 # it in the Import File for the 'aix-soname' feature, so we have
 # to replace the "-B" option with "-P" for AIX nm.
 if $NM -V 2>&1 | $GREP 'GNU' > /dev/null; then
   _LT_TAGVAR(export_symbols_cmds, $1)='$NM -Bpg $libobjs $convenience | awk '\''{ if (((\$ 2 == "T") || (\$ 2 == "D") || (\$ 2 == "B") || (\$ 2 == "W")) && ([substr](\$ 3,1,1) != ".")) { if (\$ 2 == "W") { print \$ 3 " weak" } else { print \$ 3 } } }'\'' | sort -u > $export_symbols'
 else
   _LT_TAGVAR(export_symbols_cmds, $1)='`func_echo_all $NM | $SED -e '\''s/B\([[^B]]*\)$/P\1/'\''` -PCpgl $libobjs $convenience | awk '\''{ if (((\$ 2 == "T") || (\$ 2 == "D") || (\$ 2 == "B") || (\$ 2 == "W") || (\$ 2 == "V") || (\$ 2 == "Z")) && ([substr](\$ 1,1,1) != ".")) { if ((\$ 2 == "W") || (\$ 2 == "V") || (\$ 2 == "Z")) { print \$ 1 " weak" } else { print \$ 1 } } }'\'' | sort -u > $export_symbols'
 fi
 aix_use_runtimelinking=no

 # Test if we are trying to use run time linking or normal
 # AIX style linking. If -brtl is somewhere in LDFLAGS, we
 # have runtime linking enabled, and use it for executables.
 # For shared libraries, we enable/disable runtime linking
 # depending on the kind of the shared library created -
 # when "with_aix_soname,aix_use_runtimelinking" is:
 # "aix,no"   lib.a(lib.so.V) shared, rtl:no,  for executables
 # "aix,yes"  lib.so          shared, rtl:yes, for executables
 #            lib.a           static archive
 # "both,no"  lib.so.V(shr.o) shared, rtl:yes
 #            lib.a(lib.so.V) shared, rtl:no,  for executables
 # "both,yes" lib.so.V(shr.o) shared, rtl:yes, for executables
 #            lib.a(lib.so.V) shared, rtl:no
 # "svr4,*"   lib.so.V(shr.o) shared, rtl:yes, for executables
 #            lib.a           static archive
 case $host_os in aix4.[[23]]|aix4.[[23]].*|aix[[5-9]]*)
   for ld_flag in $LDFLAGS; do
   if (test x-brtl = "x$ld_flag" || test x-Wl,-brtl = "x$ld_flag"); then
     aix_use_runtimelinking=yes
     break
   fi
   done
   if test svr4,no = "$with_aix_soname,$aix_use_runtimelinking"; then
     # With aix-soname=svr4, we create the lib.so.V shared archives only,
     # so we don't have lib.a shared libs to link our executables.
     # We have to force runtime linking in this case.
     aix_use_runtimelinking=yes
     LDFLAGS="$LDFLAGS -Wl,-brtl"
   fi
   ;;
 esac

 exp_sym_flag='-bexport'
 no_entry_flag='-bnoentry'
     fi

     # When large executables or shared objects are built, AIX ld can
     # have problems creating the table of contents.  If linking a library
     # or program results in "error TOC overflow" add -mminimal-toc to
     # CXXFLAGS/CFLAGS for g++/gcc.  In the cases where that is not
     # enough to fix the problem, add -Wl,-bbigtoc to LDFLAGS.

     _LT_TAGVAR(archive_cmds, $1)=''
     _LT_TAGVAR(hardcode_direct, $1)=yes
     _LT_TAGVAR(hardcode_direct_absolute, $1)=yes
     _LT_TAGVAR(hardcode_libdir_separator, $1)=':'
     _LT_TAGVAR(link_all_deplibs, $1)=yes
     _LT_TAGVAR(file_list_spec, $1)='$wl-f,'
     case $with_aix_soname,$aix_use_runtimelinking in
     aix,*) ;; # traditional, no import file
     svr4,* | *,yes) # use import file
 # The Import File defines what to hardcode.
 _LT_TAGVAR(hardcode_direct, $1)=no
 _LT_TAGVAR(hardcode_direct_absolute, $1)=no
 ;;
     esac

     if test yes = "$GCC"; then
 case $host_os in aix4.[[012]]|aix4.[[012]].*)
 # We only want to do this on AIX 4.2 and lower, the check
 # below for broken collect2 doesn't work under 4.3+
   collect2name=`$CC -print-prog-name=collect2`
   if test -f "$collect2name" &&
    strings "$collect2name" | $GREP resolve_lib_name >/dev/null
   then
   # We have reworked collect2
   :
   else
   # We have old collect2
   _LT_TAGVAR(hardcode_direct, $1)=unsupported
   # It fails to find uninstalled libraries when the uninstalled
   # path is not listed in the libpath.  Setting hardcode_minus_L
   # to unsupported forces relinking
   _LT_TAGVAR(hardcode_minus_L, $1)=yes
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
  _LT_TAGVAR(hardcode_libdir_separator, $1)=
  fi
  ;;
 esac
 shared_flag='-shared'
 if test yes = "$aix_use_runtimelinking"; then
  shared_flag="$shared_flag "'$wl-G'
 fi
 # Need to ensure runtime linking is disabled for the traditional
 # shared library, or the linker may eventually find shared libraries
 # /with/ Import File - we do not want to mix them.
 shared_flag_aix='-shared'
 shared_flag_svr4='-shared $wl-G'
   else
 # not using gcc
 if test ia64 = "$host_cpu"; then
 # VisualAge C++, Version 5.5 for AIX 5L for IA-64, Beta 3 Release
 # chokes on -Wl,-G. The following line is correct:
  shared_flag='-G'
 else
  if test yes = "$aix_use_runtimelinking"; then
   shared_flag='$wl-G'
  else
   shared_flag='$wl-bM:SRE'
  fi
  shared_flag_aix='$wl-bM:SRE'
  shared_flag_svr4='$wl-G'
 fi
   fi

   _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-bexpall'
   # It seems that -bexpall does not export symbols beginning with
   # underscore (_), so it is better to generate a list of symbols to export.
   _LT_TAGVAR(always_export_symbols, $1)=yes
   if test aix,yes = "$with_aix_soname,$aix_use_runtimelinking"; then
 # Warning - without using the other runtime loading flags (-brtl),
 # -berok will link without error, but may produce a broken library.
 _LT_TAGVAR(allow_undefined_flag, $1)='-berok'
    # Determine the default libpath from the value encoded in an
    # empty executable.
    _LT_SYS_MODULE_PATH_AIX([$1])
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-blibpath:$libdir:'"$aix_libpath"
    _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -o $output_objdir/$soname $libobjs $deplibs
$wl'$no_entry_flag' $compiler_flags `if test -n "$allow_undefined_flag"; then func_echo_all
"$wl$allow_undefined_flag"; else :; fi` $wl'$exp_sym_flag:\$export_symbols' '$shared_flag
   else
 if test ia64 = "$host_cpu"; then
  _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-R $libdir:/usr/lib:/lib'
```

```
   _LT_TAGVAR(allow_undefined_flag, $1)="-z nodefs"
   _LT_TAGVAR(archive_expsym_cmds, $1)="\$CC $shared_flag"' -o $output_objdir/$soname $libobjs $deplibs
'"\$wl$no_entry_flag"' $compiler_flags $wl$allow_undefined_flag '"\$wl$exp_sym_flag:\$export_symbols"
 else
  # Determine the default libpath from the value encoded in an
  # empty executable.
  _LT_SYS_MODULE_PATH_AIX([$1])
  _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-blibpath:$libdir:'"$aix_libpath"
   # Warning - without using the other run time loading flags,
   # -berok will link without error, but may produce a broken library.
   _LT_TAGVAR(no_undefined_flag, $1)=' $wl-bernotok'
   _LT_TAGVAR(allow_undefined_flag, $1)=' $wl-berok'
   if test yes = "$with_gnu_ld"; then
     # We only use this code for GNU lds that support --whole-archive.
     _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive$convenience $wl--no-whole-archive'
   else
     # Exported symbols can be pulled into shared objects from archives
     _LT_TAGVAR(whole_archive_flag_spec, $1)='$convenience'
   fi
   _LT_TAGVAR(archive_cmds_need_lc, $1)=yes
   _LT_TAGVAR(archive_expsym_cmds, $1)='$RM -r $output_objdir/$realname.d~$MKDIR
$output_objdir/$realname.d'
   # -brtl affects multiple linker settings, -berok does not and is overridden later
   compiler_flags_filtered='`func_echo_all "$compiler_flags " | $SED -e "s%-brtl\\([[, ]]\\)%-berok\\1%g"`'
   if test svr4 != "$with_aix_soname"; then
     # This is similar to how AIX traditionally builds its shared libraries.
     _LT_TAGVAR(archive_expsym_cmds, $1)="$_LT_TAGVAR(archive_expsym_cmds, $1)"'~$CC
'$shared_flag_aix' -o $output_objdir/$realname.d/$soname $libobjs $deplibs $wl-bnoentry
'$compiler_flags_filtered'$wl-bE:$export_symbols$allow_undefined_flag~$AR $AR_FLAGS
$output_objdir/$libname$release.a $output_objdir/$realname.d/$soname'
   fi
   if test aix != "$with_aix_soname"; then
     _LT_TAGVAR(archive_expsym_cmds, $1)="$_LT_TAGVAR(archive_expsym_cmds, $1)"'~$CC
'$shared_flag_svr4' -o $output_objdir/$realname.d/$shared_archive_member_spec.o $libobjs $deplibs $wl-bnoentry
'$compiler_flags_filtered'$wl-bE:$export_symbols$allow_undefined_flag~$STRIP -e
$output_objdir/$realname.d/$shared_archive_member_spec.o~( func_echo_all "#!
$soname($shared_archive_member_spec.o)"; if test shr_64 = "$shared_archive_member_spec"; then func_echo_all
"# 64"; else func_echo_all "# 32"; fi; cat $export_symbols ) >
$output_objdir/$realname.d/$shared_archive_member_spec.imp~$AR $AR_FLAGS $output_objdir/$soname
$output_objdir/$realname.d/$shared_archive_member_spec.o
$output_objdir/$realname.d/$shared_archive_member_spec.imp'
   else
     # used by -dlpreopen to get the symbols
     _LT_TAGVAR(archive_expsym_cmds, $1)="$_LT_TAGVAR(archive_expsym_cmds, $1)"'~$MV
$output_objdir/$realname.d/$soname $output_objdir'
   fi
   _LT_TAGVAR(archive_expsym_cmds, $1)="$_LT_TAGVAR(archive_expsym_cmds, $1)"'~$RM -r
$output_objdir/$realname.d'
```

```
 fi
   fi
   ;;

  amigaos*)
   case $host_cpu in
   powerpc)
      # see comment about AmigaOS4 .so support
      _LT_TAGVAR(archive_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname -o $lib'
      _LT_TAGVAR(archive_expsym_cmds, $1)="
    ;;
   m68k)
      _LT_TAGVAR(archive_cmds, $1)='$RM $output_objdir/a2ixlibrary.data~$ECHO "#define NAME
$libname" > $output_objdir/a2ixlibrary.data~$ECHO "#define LIBRARY_ID 1" >>
$output_objdir/a2ixlibrary.data~$ECHO "#define VERSION $major" >> $output_objdir/a2ixlibrary.data~$ECHO
"#define REVISION $revision" >> $output_objdir/a2ixlibrary.data~$AR $AR_FLAGS $lib $libobjs~$RANLIB
$lib~(cd $output_objdir && a2ixlibrary -32)'
      _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
      _LT_TAGVAR(hardcode_minus_L, $1)=yes
    ;;
   esac
   ;;

  bsdi[[45]]*)
   _LT_TAGVAR(export_dynamic_flag_spec, $1)=-rdynamic
   ;;

  cygwin* | mingw* | pw32* | cegcc*)
   # When not using gcc, we currently assume that we are using
   # Microsoft Visual C++.
   # hardcode_libdir_flag_spec is actually meaningless, as there is
   # no search path for DLLs.
   case $cc_basename in
   cl*)
 # Native MSVC
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)=' '
 _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
 _LT_TAGVAR(always_export_symbols, $1)=yes
 _LT_TAGVAR(file_list_spec, $1)='@'
 # Tell ltmain to make .lib files, not .a files.
 libext=lib
 # Tell ltmain to make .dll files, not .so files.
 shrext_cmds=.dll
 # FIXME: Setting linknames here is a bad hack.
 _LT_TAGVAR(archive_cmds, $1)='$CC -o $output_objdir/$soname $libobjs $compiler_flags $deplibs -Wl,-DLL,-
IMPLIB:"$tool_output_objdir$libname.dll.lib"~linknames='
 _LT_TAGVAR(archive_expsym_cmds, $1)='if _LT_DLL_DEF_P([$export_symbols]); then
```

```
          cp "$export_symbols" "$output_objdir/$soname.def";
          echo "$tool_output_objdir$soname.def" > "$output_objdir/$soname.exp";
        else
          $SED -e '\''s/^/-link -EXPORT:/'\'' < $export_symbols > $output_objdir/$soname.exp;
        fi~
        $CC -o $tool_output_objdir$soname $libobjs $compiler_flags $deplibs "@$tool_output_objdir$soname.exp" -
Wl,-DLL,-IMPLIB:"$tool_output_objdir$libname.dll.lib"~
        linknames='
 # The linker will not automatically build a static lib if we build a DLL.
 # _LT_TAGVAR(old_archive_from_new_cmds, $1)='true'
 _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=yes
 _LT_TAGVAR(exclude_expsyms, $1)='_NULL_IMPORT_DESCRIPTOR|_IMPORT_DESCRIPTOR_.*'
 _LT_TAGVAR(export_symbols_cmds, $1)='$NM $libobjs $convenience | $global_symbol_pipe | $SED -e
'\''/^[[BCDGRS]][[ ]]/s/.*[[ ]]\([[^ ]]*\)/\1,DATA/'\'' | $SED -e '\''/^[[AITW]][[ ]]/s/.*[[ ]]//'\'' | sort | uniq >
$export_symbols'
 # Don't use ranlib
 _LT_TAGVAR(old_postinstall_cmds, $1)='chmod 644 $oldlib'
 _LT_TAGVAR(postlink_cmds, $1)='lt_outputfile="@OUTPUT@"~
        lt_tool_outputfile="@TOOL_OUTPUT@"~
        case $lt_outputfile in
          *.exe|*.EXE) ;;
          *)
            lt_outputfile=$lt_outputfile.exe
            lt_tool_outputfile=$lt_tool_outputfile.exe
            ;;
        esac~
        if test : != "$MANIFEST_TOOL" && test -f "$lt_outputfile.manifest"; then
          $MANIFEST_TOOL -manifest "$lt_tool_outputfile.manifest" -outputresource:"$lt_tool_outputfile" || exit 1;
          $RM "$lt_outputfile.manifest";
        fi'
 ;;
    *)
 # Assume MSVC wrapper
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)=' '
 _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
 # Tell ltmain to make .lib files, not .a files.
 libext=lib
 # Tell ltmain to make .dll files, not .so files.
 shrext_cmds=.dll
 # FIXME: Setting linknames here is a bad hack.
 _LT_TAGVAR(archive_cmds, $1)='$CC -o $lib $libobjs $compiler_flags `func_echo_all "$deplibs" | $SED '\''s/ -
lc$//'\''` -link -dll~linknames='
 # The linker will automatically build a .lib file if we build a DLL.
 _LT_TAGVAR(old_archive_from_new_cmds, $1)='true'
 # FIXME: Should let the user specify the lib program.
 _LT_TAGVAR(old_archive_cmds, $1)='lib -OUT:$oldlib$oldobjs$old_deplibs'
 _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=yes
 ;;
```

```
    esac
    ;;

  darwin* | rhapsody*)
    _LT_DARWIN_LINKER_FEATURES($1)
    ;;

  dgux*)
    _LT_TAGVAR(archive_cmds, $1)='$LD -G -h $soname -o $lib $libobjs $deplibs $linker_flags'
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    ;;

  # FreeBSD 2.2.[012] allows us to include c++rt0.o to get C++ constructor
  # support.  Future versions do this automatically, but an explicit c++rt0.o
  # does not break anything, and helps significantly (at the cost of a little
  # extra space).
  freebsd2.2*)
    _LT_TAGVAR(archive_cmds, $1)='$LD -Bshareable -o $lib $libobjs $deplibs $linker_flags /usr/lib/c++rt0.o'
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-R$libdir'
    _LT_TAGVAR(hardcode_direct, $1)=yes
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    ;;

  # Unfortunately, older versions of FreeBSD 2 do not have this feature.
  freebsd2.*)
    _LT_TAGVAR(archive_cmds, $1)='$LD -Bshareable -o $lib $libobjs $deplibs $linker_flags'
    _LT_TAGVAR(hardcode_direct, $1)=yes
    _LT_TAGVAR(hardcode_minus_L, $1)=yes
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    ;;

  # FreeBSD 3 and greater uses gcc -shared to do shared libraries.
  freebsd* | dragonfly*)
    _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag -o $lib $libobjs $deplibs $compiler_flags'
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-R$libdir'
    _LT_TAGVAR(hardcode_direct, $1)=yes
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    ;;

  hpux9*)
    if test yes = "$GCC"; then
 _LT_TAGVAR(archive_cmds, $1)='$RM $output_objdir/$soname~$CC -shared $pic_flag $wl+b $wl$install_libdir
-o $output_objdir/$soname $libobjs $deplibs $compiler_flags~test "x$output_objdir/$soname" = "x$lib" || mv -f
$output_objdir/$soname $lib'
    else
 _LT_TAGVAR(archive_cmds, $1)='$RM $output_objdir/$soname~$LD -b +b $install_libdir -o
$output_objdir/$soname $libobjs $deplibs $linker_flags~test "x$output_objdir/$soname" = "x$lib" || mv -f
```

```
$output_objdir/$soname $lib'
     fi
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl+b $wl$libdir'
     _LT_TAGVAR(hardcode_libdir_separator, $1)=:
     _LT_TAGVAR(hardcode_direct, $1)=yes

     # hardcode_minus_L: Not really in the search PATH,
     # but as the default location of the library.
     _LT_TAGVAR(hardcode_minus_L, $1)=yes
     _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'
     ;;

   hpux10*)
     if test yes,no = "$GCC,$with_gnu_ld"; then
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $wl+h $wl$soname $wl+b $wl$install_libdir -o $lib
$libobjs $deplibs $compiler_flags'
     else
 _LT_TAGVAR(archive_cmds, $1)='$LD -b +h $soname +b $install_libdir -o $lib $libobjs $deplibs $linker_flags'
     fi
     if test no = "$with_gnu_ld"; then
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl+b $wl$libdir'
 _LT_TAGVAR(hardcode_libdir_separator, $1)=:
 _LT_TAGVAR(hardcode_direct, $1)=yes
 _LT_TAGVAR(hardcode_direct_absolute, $1)=yes
 _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'
 # hardcode_minus_L: Not really in the search PATH,
 # but as the default location of the library.
 _LT_TAGVAR(hardcode_minus_L, $1)=yes
     fi
     ;;

   hpux11*)
     if test yes,no = "$GCC,$with_gnu_ld"; then
 case $host_cpu in
 hppa*64*)
   _LT_TAGVAR(archive_cmds, $1)='$CC -shared $wl+h $wl$soname -o $lib $libobjs $deplibs $compiler_flags'
   ;;
 ia64*)
   _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $wl+h $wl$soname $wl+nodefaultrpath -o $lib
$libobjs $deplibs $compiler_flags'
   ;;
 *)
   _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $wl+h $wl$soname $wl+b $wl$install_libdir -o $lib
$libobjs $deplibs $compiler_flags'
   ;;
 esac
     else
 case $host_cpu in
 hppa*64*)
   _LT_TAGVAR(archive_cmds, $1)='$CC -b $wl+h $wl$soname -o $lib $libobjs $deplibs $compiler_flags'
   ;;
 ia64*)
   _LT_TAGVAR(archive_cmds, $1)='$CC -b $wl+h $wl$soname $wl+nodefaultrpath -o $lib $libobjs $deplibs
$compiler_flags'
   ;;
 *)
 m4_if($1, [], [
   # Older versions of the 11.00 compiler do not understand -b yet
   # (HP92453-01 A.11.01.20 doesn't, HP92453-01 B.11.X.35175-35176.GP does)
   _LT_LINKER_OPTION([if $CC understands -b],
     _LT_TAGVAR(lt_cv_prog_compiler__b, $1), [-b],
     [_LT_TAGVAR(archive_cmds, $1)='$CC -b $wl+h $wl$soname $wl+b $wl$install_libdir -o $lib $libobjs
$deplibs $compiler_flags'],
     [_LT_TAGVAR(archive_cmds, $1)='$LD -b +h $soname +b $install_libdir -o $lib $libobjs $deplibs
$linker_flags'])],
   [_LT_TAGVAR(archive_cmds, $1)='$CC -b $wl+h $wl$soname $wl+b $wl$install_libdir -o $lib $libobjs
$deplibs $compiler_flags'])
   ;;
 esac
     fi
     if test no = "$with_gnu_ld"; then
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl+b $wl$libdir'
 _LT_TAGVAR(hardcode_libdir_separator, $1)=:

 case $host_cpu in
 hppa*64*|ia64*)
   _LT_TAGVAR(hardcode_direct, $1)=no
   _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
   ;;
 *)
   _LT_TAGVAR(hardcode_direct, $1)=yes
   _LT_TAGVAR(hardcode_direct_absolute, $1)=yes
   _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'

   # hardcode_minus_L: Not really in the search PATH,
   # but as the default location of the library.
   _LT_TAGVAR(hardcode_minus_L, $1)=yes
   ;;
 esac
     fi
     ;;

   irix5* | irix6* | nonstopux*)
     if test yes = "$GCC"; then
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname `test -n "$verstring" && func_echo_all "$wl-set_version $wl$verstring"` $wl-update_registry
```

```
$wl$output_objdir/so_locations -o $lib'
 # Try to use the -exported_symbol ld option, if it does not
 # work, assume that -exports_file does not work either and
 # implicitly export all symbols.
 # This should be the same for all languages, so no per-tag cache variable.
 AC_CACHE_CHECK([whether the $host_os linker accepts -exported_symbol],
   [lt_cv_irix_exported_symbol],
   [save_LDFLAGS=$LDFLAGS
    LDFLAGS="$LDFLAGS -shared $wl-exported_symbol ${wl}foo $wl-update_registry $wl/dev/null"
    AC_LINK_IFELSE(
      [AC_LANG_SOURCE(
         [AC_LANG_CASE([C], [[int foo (void) { return 0; }]],
         [C++], [[int foo (void) { return 0; }]],
         [Fortran 77], [[
     subroutine foo
     end]],
         [Fortran], [[
     subroutine foo
     end]])])],
       [lt_cv_irix_exported_symbol=yes],
       [lt_cv_irix_exported_symbol=no])
          LDFLAGS=$save_LDFLAGS])
 if test yes = "$lt_cv_irix_exported_symbol"; then
         _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-soname $wl$soname `test -n "$verstring" && func_echo_all "$wl-set_version $wl$verstring"` $wl-update_registry $wl$output_objdir/so_locations $wl-exports_file $wl$export_symbols -o $lib'
 fi
 _LT_TAGVAR(link_all_deplibs, $1)=no
     else
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags -soname $soname `test -n "$verstring" && func_echo_all "-set_version $verstring"` -update_registry $output_objdir/so_locations -o $lib'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags -soname $soname `test -n "$verstring" && func_echo_all "-set_version $verstring"` -update_registry $output_objdir/so_locations -exports_file $export_symbols -o $lib'
     fi
     _LT_TAGVAR(archive_cmds_need_lc, $1)='no'
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
     _LT_TAGVAR(hardcode_libdir_separator, $1)=:
     _LT_TAGVAR(inherit_rpath, $1)=yes
     _LT_TAGVAR(link_all_deplibs, $1)=yes
     ;;

   linux*)
     case $cc_basename in
     tcc*)
 # Fabrice Bellard et al's Tiny C Compiler
 _LT_TAGVAR(ld_shlibs, $1)=yes
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag -o $lib $libobjs $deplibs $compiler_flags'
```

```
;;
    esac
    ;;

  netbsd* | netbsdelf*-gnu)
    if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
 _LT_TAGVAR(archive_cmds, $1)='$LD -Bshareable -o $lib $libobjs $deplibs $linker_flags'  # a.out
    else
 _LT_TAGVAR(archive_cmds, $1)='$LD -shared -o $lib $libobjs $deplibs $linker_flags'      # ELF
    fi
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-R$libdir'
    _LT_TAGVAR(hardcode_direct, $1)=yes
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    ;;

  newsos6)
    _LT_TAGVAR(archive_cmds, $1)='$LD -G -h $soname -o $lib $libobjs $deplibs $linker_flags'
    _LT_TAGVAR(hardcode_direct, $1)=yes
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
    _LT_TAGVAR(hardcode_libdir_separator, $1)=:
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    ;;

  *nto* | *qnx*)
    ;;

  openbsd* | bitrig*)
    if test -f /usr/libexec/ld.so; then
 _LT_TAGVAR(hardcode_direct, $1)=yes
 _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
 _LT_TAGVAR(hardcode_direct_absolute, $1)=yes
 if test -z "`echo __ELF__ | $CC -E - | $GREP __ELF__`"; then
   _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag -o $lib $libobjs $deplibs $compiler_flags'
   _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $pic_flag -o $lib $libobjs $deplibs $compiler_flags
$wl-retain-symbols-file,$export_symbols'
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath,$libdir'
   _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'
 else
   _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag -o $lib $libobjs $deplibs $compiler_flags'
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath,$libdir'
 fi
    else
 _LT_TAGVAR(ld_shlibs, $1)=no
    fi
    ;;

  os2*)
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
```

```
     _LT_TAGVAR(hardcode_minus_L, $1)=yes
     _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
     shrext_cmds=.dll
     _LT_TAGVAR(archive_cmds, $1)='$ECHO "LIBRARY ${soname%$shared_ext} INITINSTANCE
TERMINSTANCE" > $output_objdir/$libname.def~
 $ECHO "DESCRIPTION \"$libname\"" >> $output_objdir/$libname.def~
 $ECHO "DATA MULTIPLE NONSHARED" >> $output_objdir/$libname.def~
 $ECHO EXPORTS >> $output_objdir/$libname.def~
 emxexp $libobjs | $SED /"_DLL_InitTerm"/d >> $output_objdir/$libname.def~
 $CC -Zdll -Zcrtdll -o $output_objdir/$soname $libobjs $deplibs $compiler_flags $output_objdir/$libname.def~
 emximp -o $lib $output_objdir/$libname.def'
     _LT_TAGVAR(archive_expsym_cmds, $1)='$ECHO "LIBRARY ${soname%$shared_ext} INITINSTANCE
TERMINSTANCE" > $output_objdir/$libname.def~
 $ECHO "DESCRIPTION \"$libname\"" >> $output_objdir/$libname.def~
 $ECHO "DATA MULTIPLE NONSHARED" >> $output_objdir/$libname.def~
 $ECHO EXPORTS >> $output_objdir/$libname.def~
 prefix_cmds="$SED"~
 if test EXPORTS = "`$SED 1q $export_symbols`"; then
   prefix_cmds="$prefix_cmds -e 1d";
 fi~
 prefix_cmds="$prefix_cmds -e \"s/^\(.*\)$/_\1/g\""~
 cat $export_symbols | $prefix_cmds >> $output_objdir/$libname.def~
 $CC -Zdll -Zcrtdll -o $output_objdir/$soname $libobjs $deplibs $compiler_flags $output_objdir/$libname.def~
 emximp -o $lib $output_objdir/$libname.def'
     _LT_TAGVAR(old_archive_From_new_cmds, $1)='emximp -o $output_objdir/${libname}_dll.a
$output_objdir/$libname.def'
     _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=yes
     ;;

   osf3*)
     if test yes = "$GCC"; then
 _LT_TAGVAR(allow_undefined_flag, $1)=' $wl-expect_unresolved $wl\*'
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared$allow_undefined_flag $libobjs $deplibs $compiler_flags $wl-
soname $wl$soname `test -n "$verstring" && func_echo_all "$wl-set_version $wl$verstring"` $wl-update_registry
$wl$output_objdir/so_locations -o $lib'
     else
 _LT_TAGVAR(allow_undefined_flag, $1)=' -expect_unresolved \*'
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared$allow_undefined_flag $libobjs $deplibs $compiler_flags -soname
$soname `test -n "$verstring" && func_echo_all "-set_version $verstring"` -update_registry
$output_objdir/so_locations -o $lib'
     fi
     _LT_TAGVAR(archive_cmds_need_lc, $1)='no'
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
     _LT_TAGVAR(hardcode_libdir_separator, $1)=:
     ;;

   osf4* | osf5*) # as osf3* with the addition of -msym flag
     if test yes = "$GCC"; then
 _LT_TAGVAR(allow_undefined_flag, $1)=' $wl-expect_unresolved $wl\*'
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared$allow_undefined_flag $pic_flag $libobjs $deplibs
$compiler_flags $wl-msym $wl-soname $wl$soname `test -n "$verstring" && func_echo_all "$wl-set_version
$wl$verstring"` $wl-update_registry $wl$output_objdir/so_locations -o $lib'
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
    else
 _LT_TAGVAR(allow_undefined_flag, $1)=' -expect_unresolved \*'
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared$allow_undefined_flag $libobjs $deplibs $compiler_flags -msym -
soname $soname `test -n "$verstring" && func_echo_all "-set_version $verstring"` -update_registry
$output_objdir/so_locations -o $lib'
 _LT_TAGVAR(archive_expsym_cmds, $1)='for i in `cat $export_symbols`; do printf "%s %s\\n" -exported_symbol
"\$i" >> $lib.exp; done; printf "%s\\n" "-hidden">> $lib.exp~
        $CC -shared$allow_undefined_flag $wl-input $wl$lib.exp $compiler_flags $libobjs $deplibs -soname
$soname `test -n "$verstring" && $ECHO "-set_version $verstring"` -update_registry $output_objdir/so_locations -
o $lib~$RM $lib.exp'

 # Both c and cxx compiler support -rpath directly
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-rpath $libdir'
    fi
    _LT_TAGVAR(archive_cmds_need_lc, $1)='no'
    _LT_TAGVAR(hardcode_libdir_separator, $1)=:
    ;;

  solaris*)
    _LT_TAGVAR(no_undefined_flag, $1)=' -z defs'
    if test yes = "$GCC"; then
 wlarc='$wl'
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $wl-z ${wl}text $wl-h $wl$soname -o $lib $libobjs
$deplibs $compiler_flags'
 _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $lib.exp~cat $export_symbols | $SED -e
"s/\(.*\)/\1;/" >> $lib.exp~echo "local: *; };" >> $lib.exp~
        $CC -shared $pic_flag $wl-z ${wl}text $wl-M $wl$lib.exp $wl-h $wl$soname -o $lib $libobjs $deplibs
$compiler_flags~$RM $lib.exp'
    else
 case `$CC -V 2>&1` in
 *"Compilers 5.0"*)
  wlarc=''
  _LT_TAGVAR(archive_cmds, $1)='$LD -G$allow_undefined_flag -h $soname -o $lib $libobjs $deplibs
$linker_flags'
  _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $lib.exp~cat $export_symbols | $SED -e
"s/\(.*\)/\1;/" >> $lib.exp~echo "local: *; };" >> $lib.exp~
          $LD -G$allow_undefined_flag -M $lib.exp -h $soname -o $lib $libobjs $deplibs $linker_flags~$RM
$lib.exp'
  ;;
 *)
  wlarc='$wl'
  _LT_TAGVAR(archive_cmds, $1)='$CC -G$allow_undefined_flag -h $soname -o $lib $libobjs $deplibs
$compiler_flags'
```

```
   _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $lib.exp~cat $export_symbols | $SED -e
"s/\(.*\)/\1;/" >> $lib.exp~echo "local: *; };" >> $lib.exp~
           $CC -G$allow_undefined_flag -M $lib.exp -h $soname -o $lib $libobjs $deplibs $compiler_flags~$RM
$lib.exp'
   ;;
 esac
     fi
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-R$libdir'
     _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
     case $host_os in
     solaris2.[[0-5]] | solaris2.[[0-5]].*) ;;
     *)
 # The compiler driver will combine and reorder linker options,
 # but understands '-z linker_flag'.  GCC discards it without '$wl',
 # but is careful enough not to reorder.
 # Supported since Solaris 2.6 (maybe 2.5.1?)
 if test yes = "$GCC"; then
   _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl-z ${wl}allextract$convenience $wl-z ${wl}defaultextract'
 else
   _LT_TAGVAR(whole_archive_flag_spec, $1)='-z allextract$convenience -z defaultextract'
 fi
 ;;
     esac
     _LT_TAGVAR(link_all_deplibs, $1)=yes
     ;;

   sunos4*)
     if test sequent = "$host_vendor"; then
 # Use $CC to link under sequent, because it throws in some extra .o
 # files that make .init and .fini sections work.
 _LT_TAGVAR(archive_cmds, $1)='$CC -G $wl-h $soname -o $lib $libobjs $deplibs $compiler_flags'
     else
 _LT_TAGVAR(archive_cmds, $1)='$LD -assert pure-text -Bstatic -o $lib $libobjs $deplibs $linker_flags'
     fi
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
     _LT_TAGVAR(hardcode_direct, $1)=yes
     _LT_TAGVAR(hardcode_minus_L, $1)=yes
     _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
     ;;

   sysv4)
     case $host_vendor in
 sni)
   _LT_TAGVAR(archive_cmds, $1)='$LD -G -h $soname -o $lib $libobjs $deplibs $linker_flags'
   _LT_TAGVAR(hardcode_direct, $1)=yes # is this really true???
 ;;
 siemens)
   ## LD is ld it makes a PLAMLIB
```

```
  ## CC just makes a GrossModule.
  _LT_TAGVAR(archive_cmds, $1)='$LD -G -o $lib $libobjs $deplibs $linker_flags'
  _LT_TAGVAR(reload_cmds, $1)='$CC -r -o $output$reload_objs'
  _LT_TAGVAR(hardcode_direct, $1)=no
      ;;
 motorola)
  _LT_TAGVAR(archive_cmds, $1)='$LD -G -h $soname -o $lib $libobjs $deplibs $linker_flags'
  _LT_TAGVAR(hardcode_direct, $1)=no #Motorola manual says yes, but my tests say they lie
 ;;
    esac
    runpath_var='LD_RUN_PATH'
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    ;;

  sysv4.3*)
    _LT_TAGVAR(archive_cmds, $1)='$LD -G -h $soname -o $lib $libobjs $deplibs $linker_flags'
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    _LT_TAGVAR(export_dynamic_flag_spec, $1)='-Bexport'
    ;;

  sysv4*MP*)
    if test -d /usr/nec; then
 _LT_TAGVAR(archive_cmds, $1)='$LD -G -h $soname -o $lib $libobjs $deplibs $linker_flags'
 _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
 runpath_var=LD_RUN_PATH
 hardcode_runpath_var=yes
 _LT_TAGVAR(ld_shlibs, $1)=yes
    fi
    ;;

  sysv4*uw2* | sysv5OpenUNIX* | sysv5UnixWare7.[[01]].[[10]]* | unixware7* | sco3.2v5.0.[[024]]*)
    _LT_TAGVAR(no_undefined_flag, $1)='$wl-z,text'
    _LT_TAGVAR(archive_cmds_need_lc, $1)=no
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    runpath_var='LD_RUN_PATH'

    if test yes = "$GCC"; then
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $wl-Bexport:$export_symbols $wl-h,$soname -o $lib
$libobjs $deplibs $compiler_flags'
    else
 _LT_TAGVAR(archive_cmds, $1)='$CC -G $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -G $wl-Bexport:$export_symbols $wl-h,$soname -o $lib
$libobjs $deplibs $compiler_flags'
    fi
    ;;

  sysv5* | sco3.2v5* | sco5v6*)
	# Note: We CANNOT use -z defs as we might desire, because we do not
	# link with -lc, and that would cause any symbols used from libc to
	# always be unresolved, which means just about no library would
	# ever link correctly.  If we're not using GNU ld we use -z text
	# though, which does catch some bad symbols but isn't as heavy-handed
	# as -z defs.
	_LT_TAGVAR(no_undefined_flag, $1)='$wl-z,text'
	_LT_TAGVAR(allow_undefined_flag, $1)='$wl-z,nodefs'
	_LT_TAGVAR(archive_cmds_need_lc, $1)=no
	_LT_TAGVAR(hardcode_shlibpath_var, $1)=no
	_LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-R,$libdir'
	_LT_TAGVAR(hardcode_libdir_separator, $1)=':'
	_LT_TAGVAR(link_all_deplibs, $1)=yes
	_LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-Bexport'
	runpath_var='LD_RUN_PATH'

	if test yes = "$GCC"; then
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $wl-Bexport:$export_symbols $wl-h,$soname -o $lib
$libobjs $deplibs $compiler_flags'
	else
 _LT_TAGVAR(archive_cmds, $1)='$CC -G $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -G $wl-Bexport:$export_symbols $wl-h,$soname -o $lib
$libobjs $deplibs $compiler_flags'
	fi
	;;

  uts4*)
	_LT_TAGVAR(archive_cmds, $1)='$LD -G -h $soname -o $lib $libobjs $deplibs $linker_flags'
	_LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
	_LT_TAGVAR(hardcode_shlibpath_var, $1)=no
	;;

  *)
	_LT_TAGVAR(ld_shlibs, $1)=no
	;;
  esac

  if test sni = "$host_vendor"; then
	case $host in
	sysv4 | sysv4.2uw2* | sysv4.3* | sysv5*)
 _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-Blargedynsym'
 ;;
	esac
  fi
 fi
])
AC_MSG_RESULT([$_LT_TAGVAR(ld_shlibs, $1)])
```

```
test no = "$_LT_TAGVAR(ld_shlibs, $1)" && can_build_shared=no

_LT_TAGVAR(with_gnu_ld, $1)=$with_gnu_ld

_LT_DECL([], [libext], [0], [Old archive suffix (normally "a")])dnl
_LT_DECL([], [shrext_cmds], [1], [Shared library suffix (normally ".so")])dnl
_LT_DECL([], [extract_expsyms_cmds], [2],
   [The commands to extract the exported symbol list from a shared archive])

#
# Do we need to explicitly link libc?
#
case "x$_LT_TAGVAR(archive_cmds_need_lc, $1)" in
x|xyes)
 # Assume -lc should be added
 _LT_TAGVAR(archive_cmds_need_lc, $1)=yes

 if test yes,yes = "$GCC,$enable_shared"; then
   case $_LT_TAGVAR(archive_cmds, $1) in
   *'~'*)
     # FIXME: we may have to deal with multi-command sequences.
     ;;
   '$CC '*)
     # Test whether the compiler implicitly links with -lc since on some
     # systems, -lgcc has to come before -lc. If gcc already passes -lc
     # to ld, don't add -lc before -lgcc.
     AC_CACHE_CHECK([whether -lc should be explicitly linked in],
 [lt_cv_]_LT_TAGVAR(archive_cmds_need_lc, $1),
 [$RM conftest*
 echo "$lt_simple_compile_test_code" > conftest.$ac_ext

 if AC_TRY_EVAL(ac_compile) 2>conftest.err; then
   soname=conftest
   lib=conftest
   libobjs=conftest.$ac_objext
   deplibs=
   wl=$_LT_TAGVAR(lt_prog_compiler_wl, $1)
   pic_flag=$_LT_TAGVAR(lt_prog_compiler_pic, $1)
   compiler_flags=-v
   linker_flags=-v
   verstring=
   output_objdir=.
   libname=conftest
   lt_save_allow_undefined_flag=$_LT_TAGVAR(allow_undefined_flag, $1)
   _LT_TAGVAR(allow_undefined_flag, $1)=
   if AC_TRY_EVAL(_LT_TAGVAR(archive_cmds, $1) 2\>\&1 \| $GREP \" -lc \" \>/dev/null 2\>\&1)
   then
     lt_cv_[]_LT_TAGVAR(archive_cmds_need_lc, $1)=no
   else
     lt_cv_[]_LT_TAGVAR(archive_cmds_need_lc, $1)=yes
   fi
   _LT_TAGVAR(allow_undefined_flag, $1)=$lt_save_allow_undefined_flag
 else
   cat conftest.err 1>&5
 fi
 $RM conftest*
 ])
     _LT_TAGVAR(archive_cmds_need_lc, $1)=$lt_cv_[]_LT_TAGVAR(archive_cmds_need_lc, $1)
     ;;
   esac
 fi
 ;;
esac

_LT_TAGDECL([build_libtool_need_lc], [archive_cmds_need_lc], [0],
   [Whether or not to add -lc for building shared libraries])
_LT_TAGDECL([allow_libtool_libs_with_static_runtimes],
   [enable_shared_with_static_runtimes], [0],
   [Whether or not to disallow shared libs when runtime libs are static])
_LT_TAGDECL([], [export_dynamic_flag_spec], [1],
   [Compiler flag to allow reflexive dlopens])
_LT_TAGDECL([], [whole_archive_flag_spec], [1],
   [Compiler flag to generate shared objects directly from archives])
_LT_TAGDECL([], [compiler_needs_object], [1],
   [Whether the compiler copes with passing no objects directly])
_LT_TAGDECL([], [old_archive_from_new_cmds], [2],
   [Create an old-style archive from a shared archive])
_LT_TAGDECL([], [old_archive_from_expsyms_cmds], [2],
   [Create a temporary old-style archive to link instead of a shared archive])
_LT_TAGDECL([], [archive_cmds], [2], [Commands used to build a shared archive])
_LT_TAGDECL([], [archive_expsym_cmds], [2])
_LT_TAGDECL([], [module_cmds], [2],
   [Commands used to build a loadable module if different from building
   a shared archive.])
_LT_TAGDECL([], [module_expsym_cmds], [2])
_LT_TAGDECL([], [with_gnu_ld], [1],
   [Whether we are building with GNU ld or not])
_LT_TAGDECL([], [allow_undefined_flag], [1],
   [Flag that allows shared libraries with undefined symbols to be built])
_LT_TAGDECL([], [no_undefined_flag], [1],
   [Flag that enforces no undefined symbols])
_LT_TAGDECL([], [hardcode_libdir_flag_spec], [1],
   [Flag to hardcode $libdir into a binary during linking.
   This must work even if $libdir does not exist])
_LT_TAGDECL([], [hardcode_libdir_separator], [1],
   [Whether we need a single "-rpath" flag with a separated argument])
```

```
_LT_TAGDECL([], [hardcode_direct], [0],
  [Set to "yes" if using DIR/libNAME$shared_ext during linking hardcodes
  DIR into the resulting binary])
_LT_TAGDECL([], [hardcode_direct_absolute], [0],
  [Set to "yes" if using DIR/libNAME$shared_ext during linking hardcodes
  DIR into the resulting binary and the resulting library dependency is
  "absolute", i.e impossible to change by setting $shlibpath_var if the
  library is relocated])
_LT_TAGDECL([], [hardcode_minus_L], [0],
  [Set to "yes" if using the -LDIR flag during linking hardcodes DIR
  into the resulting binary])
_LT_TAGDECL([], [hardcode_shlibpath_var], [0],
  [Set to "yes" if using SHLIBPATH_VAR=DIR during linking hardcodes DIR
  into the resulting binary])
_LT_TAGDECL([], [hardcode_automatic], [0],
  [Set to "yes" if building a shared library automatically hardcodes DIR
  into the library and all subsequent libraries and executables linked
  against it])
_LT_TAGDECL([], [inherit_rpath], [0],
  [Set to yes if linker adds runtime paths of dependent libraries
  to runtime path list])
_LT_TAGDECL([], [link_all_deplibs], [0],
  [Whether libtool must link a program against all its dependency libraries])
_LT_TAGDECL([], [always_export_symbols], [0],
  [Set to "yes" if exported symbols are required])
_LT_TAGDECL([], [export_symbols_cmds], [2],
  [The commands to list exported symbols])
_LT_TAGDECL([], [exclude_expsyms], [1],
  [Symbols that should not be listed in the preloaded symbols])
_LT_TAGDECL([], [include_expsyms], [1],
  [Symbols that must always be exported])
_LT_TAGDECL([], [prelink_cmds], [2],
  [Commands necessary for linking programs (against libraries) with templates])
_LT_TAGDECL([], [postlink_cmds], [2],
  [Commands necessary for finishing linking programs])
_LT_TAGDECL([], [file_list_spec], [1],
  [Specify filename containing input files])
dnl FIXME: Not yet implemented
dnl _LT_TAGDECL([], [thread_safe_flag_spec], [1],
dnl    [Compiler flag to generate thread safe objects])
])# _LT_LINKER_SHLIBS


# _LT_LANG_C_CONFIG([TAG])
# ------------------------
# Ensure that the configuration variables for a C compiler are suitably
# defined.  These variables are subsequently used by _LT_CONFIG to write
# the compiler configuration to 'libtool'.
```

```
m4_defun([_LT_LANG_C_CONFIG],
[m4_require([_LT_DECL_EGREP])dnl
lt_save_CC=$CC
AC_LANG_PUSH(C)

# Source file extension for C test sources.
ac_ext=c

# Object file extension for compiled C test sources.
objext=o
_LT_TAGVAR(objext, $1)=$objext

# Code to be used in simple compile tests
lt_simple_compile_test_code="int some_variable = 0;"

# Code to be used in simple link tests
lt_simple_link_test_code='int main(){return(0);}'

_LT_TAG_COMPILER
# Save the default compiler, since it gets overwritten when the other
# tags are being tested, and _LT_TAGVAR(compiler, []) is a NOP.
compiler_DEFAULT=$CC

# save warnings/boilerplate of simple test code
_LT_COMPILER_BOILERPLATE
_LT_LINKER_BOILERPLATE

## CAVEAT EMPTOR:
## There is no encapsulation within the following macros, do not change
## the running order or otherwise move them around unless you know exactly
## what you are doing...
if test -n "$compiler"; then
 _LT_COMPILER_NO_RTTI($1)
 _LT_COMPILER_PIC($1)
 _LT_COMPILER_C_O($1)
 _LT_COMPILER_FILE_LOCKS($1)
 _LT_LINKER_SHLIBS($1)
 _LT_SYS_DYNAMIC_LINKER($1)
 _LT_LINKER_HARDCODE_LIBPATH($1)
 LT_SYS_DLOPEN_SELF
 _LT_CMD_STRIPLIB

 # Report what library types will actually be built
 AC_MSG_CHECKING([if libtool supports shared libraries])
 AC_MSG_RESULT([$can_build_shared])

 AC_MSG_CHECKING([whether to build shared libraries])
 test no = "$can_build_shared" && enable_shared=no
 # On AIX, shared libraries and static libraries use the same namespace, and
 # are all built from PIC.
 case $host_os in
 aix3*)
   test yes = "$enable_shared" && enable_static=no
   if test -n "$RANLIB"; then
     archive_cmds="$archive_cmds~\$RANLIB \$lib"
     postinstall_cmds='$RANLIB $lib'
   fi
   ;;

 aix[[4-9]]*)
   if test ia64 != "$host_cpu"; then
     case $enable_shared,$with_aix_soname,$aix_use_runtimelinking in
     yes,aix,yes) ;;   # shared object as lib.so file only
     yes,svr4,*) ;;   # shared object as lib.so archive member only
     yes,*) enable_static=no ;; # shared object in lib.a archive as well
     esac
   fi
   ;;
 esac
 AC_MSG_RESULT([$enable_shared])

 AC_MSG_CHECKING([whether to build static libraries])
 # Make sure either enable_shared or enable_static is yes.
 test yes = "$enable_shared" || enable_static=yes
 AC_MSG_RESULT([$enable_static])

 _LT_CONFIG($1)
fi
AC_LANG_POP
CC=$lt_save_CC
])# _LT_LANG_C_CONFIG


# _LT_LANG_CXX_CONFIG([TAG])
# --------------------------
# Ensure that the configuration variables for a C++ compiler are suitably
# defined.  These variables are subsequently used by _LT_CONFIG to write
# the compiler configuration to 'libtool'.
m4_defun([_LT_LANG_CXX_CONFIG],
[m4_require([_LT_FILEUTILS_DEFAULTS])dnl
m4_require([_LT_DECL_EGREP])dnl
m4_require([_LT_PATH_MANIFEST_TOOL])dnl
if test -n "$CXX" && ( test no != "$CXX" &&
   ( (test g++ = "$CXX" && `g++ -v >/dev/null 2>&1` ) ||
   (test g++ != "$CXX"))); then
 AC_PROG_CXXCPP
else
 _lt_caught_CXX_error=yes
fi

AC_LANG_PUSH(C++)
_LT_TAGVAR(archive_cmds_need_lc, $1)=no
_LT_TAGVAR(allow_undefined_flag, $1)=
_LT_TAGVAR(always_export_symbols, $1)=no
_LT_TAGVAR(archive_expsym_cmds, $1)=
_LT_TAGVAR(compiler_needs_object, $1)=no
_LT_TAGVAR(export_dynamic_flag_spec, $1)=
_LT_TAGVAR(hardcode_direct, $1)=no
_LT_TAGVAR(hardcode_direct_absolute, $1)=no
_LT_TAGVAR(hardcode_libdir_flag_spec, $1)=
_LT_TAGVAR(hardcode_libdir_separator, $1)=
_LT_TAGVAR(hardcode_minus_L, $1)=no
_LT_TAGVAR(hardcode_shlibpath_var, $1)=unsupported
_LT_TAGVAR(hardcode_automatic, $1)=no
_LT_TAGVAR(inherit_rpath, $1)=no
_LT_TAGVAR(module_cmds, $1)=
_LT_TAGVAR(module_expsym_cmds, $1)=
_LT_TAGVAR(link_all_deplibs, $1)=unknown
_LT_TAGVAR(old_archive_cmds, $1)=$old_archive_cmds
_LT_TAGVAR(reload_flag, $1)=$reload_flag
_LT_TAGVAR(reload_cmds, $1)=$reload_cmds
_LT_TAGVAR(no_undefined_flag, $1)=
_LT_TAGVAR(whole_archive_flag_spec, $1)=
_LT_TAGVAR(enable_shared_with_static_runtimes, $1)=no

# Source file extension for C++ test sources.
ac_ext=cpp

# Object file extension for compiled C++ test sources.
objext=o
_LT_TAGVAR(objext, $1)=$objext

# No sense in running all these tests if we already determined that
# the CXX compiler isn't working.  Some variables (like enable_shared)
# are currently assumed to apply to all compilers on this platform,
# and will be corrupted by setting them based on a non-working compiler.
if test yes != "$_lt_caught_CXX_error"; then
 # Code to be used in simple compile tests
 lt_simple_compile_test_code="int some_variable = 0;"

 # Code to be used in simple link tests
 lt_simple_link_test_code='int main(int, char *[[]]) { return(0); }'
```

```
# ltmain only uses $CC for tagged configurations so make sure $CC is set.
_LT_TAG_COMPILER

# save warnings/boilerplate of simple test code
_LT_COMPILER_BOILERPLATE
_LT_LINKER_BOILERPLATE

# Allow CC to be a program name with arguments.
lt_save_CC=$CC
lt_save_CFLAGS=$CFLAGS
lt_save_LD=$LD
lt_save_GCC=$GCC
GCC=$GXX
lt_save_with_gnu_ld=$with_gnu_ld
lt_save_path_LD=$lt_cv_path_LD
if test -n "${lt_cv_prog_gnu_ldcxx+set}"; then
  lt_cv_prog_gnu_ld=$lt_cv_prog_gnu_ldcxx
else
  $as_unset lt_cv_prog_gnu_ld
fi
if test -n "${lt_cv_path_LDCXX+set}"; then
  lt_cv_path_LD=$lt_cv_path_LDCXX
else
  $as_unset lt_cv_path_LD
fi
test -z "${LDCXX+set}" || LD=$LDCXX
CC=${CXX-"c++"}
CFLAGS=$CXXFLAGS
compiler=$CC
_LT_TAGVAR(compiler, $1)=$CC
_LT_CC_BASENAME([$compiler])

if test -n "$compiler"; then
  # We don't want -fno-exception when compiling C++ code, so set the
  # no_builtin_flag separately
  if test yes = "$GXX"; then
    _LT_TAGVAR(lt_prog_compiler_no_builtin_flag, $1)=' -fno-builtin'
  else
    _LT_TAGVAR(lt_prog_compiler_no_builtin_flag, $1)=
  fi

  if test yes = "$GXX"; then
    # Set up default GNU C++ configuration

    LT_PATH_LD

    # Check if GNU C++ uses GNU ld as the underlying linker, since the
    # archiving commands below assume that GNU ld is being used.
    if test yes = "$with_gnu_ld"; then
      _LT_TAGVAR(archive_cmds, $1)='$CC $pic_flag -shared -nostdlib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags $wl-soname $wl$soname -o $lib'
      _LT_TAGVAR(archive_expsym_cmds, $1)='$CC $pic_flag -shared -nostdlib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags $wl-soname $wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'

      _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
      _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl--export-dynamic'

      # If archive_cmds runs LD, not CC, wlarc should be empty
      # XXX I think wlarc can be eliminated in ltcf-cxx, but I need to
      #     investigate it a little bit more. (MM)
      wlarc='$wl'

      # ancient GNU ld didn't support --whole-archive et. al.
      if eval "`$CC -print-prog-name=ld` --help 2>&1" |
   $GREP 'no-whole-archive' > /dev/null; then
        _LT_TAGVAR(whole_archive_flag_spec, $1)=$wlarc'--whole-archive$convenience '$wlarc'--no-whole-archive'
      else
        _LT_TAGVAR(whole_archive_flag_spec, $1)=
      fi
    else
      with_gnu_ld=no
      wlarc=

      # A generic and very simple default shared library creation
      # command for GNU C++ for the case where it uses the native
      # linker, instead of GNU ld.  If possible, this setting should
      # overridden to take advantage of the native linker features on
      # the platform it is being used on.
      _LT_TAGVAR(archive_cmds, $1)='$CC -shared -nostdlib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags -o $lib'
    fi

    # Commands to make compiler produce verbose output that lists
    # what "hidden" libraries, object files and flags are used when
    # linking a shared library.
    output_verbose_link_cmd='$CC -shared $CFLAGS -v conftest.$objext 2>&1 | $GREP -v "^Configured with:" | $GREP "\-L"'

  else
    GXX=no
    with_gnu_ld=no
    wlarc=
  fi
```

```
# PORTME: fill in a description of your system's C++ link characteristics
AC_MSG_CHECKING([whether the $compiler linker ($LD) supports shared libraries])
_LT_TAGVAR(ld_shlibs, $1)=yes
case $host_os in
  aix3*)
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
  aix[[4-9]]*)
    if test ia64 = "$host_cpu"; then
      # On IA64, the linker does run time linking by default, so we don't
      # have to do anything special.
      aix_use_runtimelinking=no
      exp_sym_flag='-Bexport'
      no_entry_flag=
    else
      aix_use_runtimelinking=no

      # Test if we are trying to use run time linking or normal
      # AIX style linking. If -brtl is somewhere in LDFLAGS, we
      # have runtime linking enabled, and use it for executables.
      # For shared libraries, we enable/disable runtime linking
      # depending on the kind of the shared library created -
      # when "with_aix_soname,aix_use_runtimelinking" is:
      # "aix,no"   lib.a(lib.so.V) shared, rtl:no,  for executables
      # "aix,yes"  lib.so          shared, rtl:yes, for executables
      #            lib.a           static archive
      # "both,no"  lib.so.V(shr.o) shared, rtl:yes
      #            lib.a(lib.so.V) shared, rtl:no,  for executables
      # "both,yes" lib.so.V(shr.o) shared, rtl:yes, for executables
      #            lib.a(lib.so.V) shared, rtl:no
      # "svr4,*"   lib.so.V(shr.o) shared, rtl:yes, for executables
      #            lib.a           static archive
      case $host_os in aix4.[[23]]|aix4.[[23]].*|aix[[5-9]]*)
  for ld_flag in $LDFLAGS; do
    case $ld_flag in
    *-brtl*)
      aix_use_runtimelinking=yes
      break
      ;;
    esac
  done
  if test svr4,no = "$with_aix_soname,$aix_use_runtimelinking"; then
    # With aix-soname=svr4, we create the lib.so.V shared archives only,
    # so we don't have lib.a shared libs to link our executables.
    # We have to force runtime linking in this case.
    aix_use_runtimelinking=yes
    LDFLAGS="$LDFLAGS -Wl,-brtl"
  fi
  ;;
      esac

      exp_sym_flag='-bexport'
      no_entry_flag='-bnoentry'
    fi

    # When large executables or shared objects are built, AIX ld can
    # have problems creating the table of contents.  If linking a library
    # or program results in "error TOC overflow" add -mminimal-toc to
    # CXXFLAGS/CFLAGS for g++/gcc.  In the cases where that is not
    # enough to fix the problem, add -Wl,-bbigtoc to LDFLAGS.

    _LT_TAGVAR(archive_cmds, $1)=''
    _LT_TAGVAR(hardcode_direct, $1)=yes
    _LT_TAGVAR(hardcode_direct_absolute, $1)=yes
    _LT_TAGVAR(hardcode_libdir_separator, $1)=':'
    _LT_TAGVAR(link_all_deplibs, $1)=yes
    _LT_TAGVAR(file_list_spec, $1)='$wl-f,'
    case $with_aix_soname,$aix_use_runtimelinking in
    aix,*) ;; # no import file
    svr4,* | *,yes) # use import file
      # The Import File defines what to hardcode.
      _LT_TAGVAR(hardcode_direct, $1)=no
      _LT_TAGVAR(hardcode_direct_absolute, $1)=no
      ;;
    esac

    if test yes = "$GXX"; then
      case $host_os in aix4.[[012]]|aix4.[[012]].*)
      # We only want to do this on AIX 4.2 and lower, the check
      # below for broken collect2 doesn't work under 4.3+
collect2name=`$CC -print-prog-name=collect2`
if test -f "$collect2name" &&
    strings "$collect2name" | $GREP resolve_lib_name >/dev/null
then
  # We have reworked collect2
  :
else
  # We have old collect2
  _LT_TAGVAR(hardcode_direct, $1)=unsupported
  # It fails to find uninstalled libraries when the uninstalled
  # path is not listed in the libpath.  Setting hardcode_minus_L
  # to unsupported forces relinking
  _LT_TAGVAR(hardcode_minus_L, $1)=yes
  _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
  _LT_TAGVAR(hardcode_libdir_separator, $1)=
  fi
     esac
     shared_flag='-shared'
  if test yes = "$aix_use_runtimelinking"; then
   shared_flag=$shared_flag' $wl-G'
  fi
  # Need to ensure runtime linking is disabled for the traditional
  # shared library, or the linker may eventually find shared libraries
  # /with/ Import File - we do not want to mix them.
  shared_flag_aix='-shared'
  shared_flag_svr4='-shared $wl-G'
    else
     # not using gcc
     if test ia64 = "$host_cpu"; then
  # VisualAge C++, Version 5.5 for AIX 5L for IA-64, Beta 3 Release
  # chokes on -Wl,-G. The following line is correct:
  shared_flag='-G'
     else
   if test yes = "$aix_use_runtimelinking"; then
    shared_flag='$wl-G'
   else
    shared_flag='$wl-bM:SRE'
   fi
   shared_flag_aix='$wl-bM:SRE'
   shared_flag_svr4='$wl-G'
     fi
    fi

    _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-bexpall'
    # It seems that -bexpall does not export symbols beginning with
    # underscore (_), so it is better to generate a list of symbols to
# export.
    _LT_TAGVAR(always_export_symbols, $1)=yes
if test aix,yes = "$with_aix_soname,$aix_use_runtimelinking"; then
     # Warning - without using the other runtime loading flags (-brtl),
     # -berok will link without error, but may produce a broken library.
     # The "-G" linker flag allows undefined symbols.
     _LT_TAGVAR(no_undefined_flag, $1)='-bernotok'
     # Determine the default libpath from the value encoded in an empty
     # executable.
     _LT_SYS_MODULE_PATH_AIX([$1])
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-blibpath:$libdir:'"$aix_libpath"

     _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -o $output_objdir/$soname $libobjs $deplibs
$wl'$no_entry_flag' $compiler_flags `if test -n "$allow_undefined_flag"; then func_echo_all
"$wl$allow_undefined_flag"; else :; fi` $wl'$exp_sym_flag:\$export_symbols' '$shared_flag
    else
     if test ia64 = "$host_cpu"; then
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-R $libdir:/usr/lib:/lib'
   _LT_TAGVAR(allow_undefined_flag, $1)="-z nodefs"
   _LT_TAGVAR(archive_expsym_cmds, $1)="\$CC $shared_flag"' -o $output_objdir/$soname $libobjs $deplibs
'"\$wl$no_entry_flag"' $compiler_flags $wl$allow_undefined_flag '"\$wl$exp_sym_flag:\$export_symbols"
     else
   # Determine the default libpath from the value encoded in an
   # empty executable.
   _LT_SYS_MODULE_PATH_AIX([$1])
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-blibpath:$libdir:'"$aix_libpath"
   # Warning - without using the other run time loading flags,
   # -berok will link without error, but may produce a broken library.
   _LT_TAGVAR(no_undefined_flag, $1)=' $wl-bernotok'
   _LT_TAGVAR(allow_undefined_flag, $1)=' $wl-berok'
   if test yes = "$with_gnu_ld"; then
    # We only use this code for GNU lds that support --whole-archive.
    _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive$convenience $wl--no-whole-archive'
   else
    # Exported symbols can be pulled into shared objects from archives
    _LT_TAGVAR(whole_archive_flag_spec, $1)='$convenience'
   fi
   _LT_TAGVAR(archive_cmds_need_lc, $1)=yes
   _LT_TAGVAR(archive_expsym_cmds, $1)='$RM -r $output_objdir/$realname.d~$MKDIR
$output_objdir/$realname.d'
   # -brtl affects multiple linker settings, -berok does not and is overridden later
   compiler_flags_filtered='`func_echo_all "$compiler_flags " | $SED -e "s%-brtl\\([[, ]]\\)%-berok\\1%g"`'
   if test svr4 != "$with_aix_soname"; then
    # This is similar to how AIX traditionally builds its shared
    # libraries. Need -bnortl late, we may have -brtl in LDFLAGS.
    _LT_TAGVAR(archive_expsym_cmds, $1)="$_LT_TAGVAR(archive_expsym_cmds, $1)"'~$CC
'$shared_flag_aix' -o $output_objdir/$realname.d/$soname $libobjs $deplibs $wl-bnoentry
'$compiler_flags_filtered'$wl-bE:$export_symbols$allow_undefined_flag~$AR $AR_FLAGS
$output_objdir/$libname$release.a $output_objdir/$realname.d/$soname'
   fi
   if test aix != "$with_aix_soname"; then
    _LT_TAGVAR(archive_expsym_cmds, $1)="$_LT_TAGVAR(archive_expsym_cmds, $1)"'~$CC
'$shared_flag_svr4' -o $output_objdir/$realname.d/$shared_archive_member_spec.o $libobjs $deplibs $wl-bnoentry
'$compiler_flags_filtered'$wl-bE:$export_symbols$allow_undefined_flag~$STRIP -e
$output_objdir/$realname.d/$shared_archive_member_spec.o~( func_echo_all "#!
$soname($shared_archive_member_spec.o)"; if test shr_64 = "$shared_archive_member_spec"; then func_echo_all
"# 64"; else func_echo_all "# 32"; fi; cat $export_symbols ) >
$output_objdir/$realname.d/$shared_archive_member_spec.imp~$AR $AR_FLAGS $output_objdir/$soname
$output_objdir/$realname.d/$shared_archive_member_spec.o
$output_objdir/$realname.d/$shared_archive_member_spec.imp'
   else
    # used by -dlpreopen to get the symbols
    _LT_TAGVAR(archive_expsym_cmds, $1)="$_LT_TAGVAR(archive_expsym_cmds, $1)"'~$MV
$output_objdir/$realname.d/$soname $output_objdir'
   fi
```

```
     _LT_TAGVAR(archive_expsym_cmds, $1)="$_LT_TAGVAR(archive_expsym_cmds, $1)"'~$RM -r
$output_objdir/$realname.d'
         fi
       fi
       ;;

     beos*)
 if $LD --help 2>&1 | $GREP ': supported targets:.* elf' > /dev/null; then
   _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
   # Joseph Beckenbach <jrb3@best.com> says some releases of gcc
   # support --undefined.  This deserves some investigation.  FIXME
   _LT_TAGVAR(archive_cmds, $1)='$CC -nostart $libobjs $deplibs $compiler_flags $wl-soname $wl$soname -o
$lib'
 else
   _LT_TAGVAR(ld_shlibs, $1)=no
 fi
 ;;

     chorus*)
       case $cc_basename in
         *)
   # FIXME: insert proper C++ library support
   _LT_TAGVAR(ld_shlibs, $1)=no
   ;;
       esac
       ;;

     cygwin* | mingw* | pw32* | cegcc*)
 case $GXX,$cc_basename in
 ,cl* | no,cl*)
   # Native MSVC
   # hardcode_libdir_flag_spec is actually meaningless, as there is
   # no search path for DLLs.
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)=' '
   _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
   _LT_TAGVAR(always_export_symbols, $1)=yes
   _LT_TAGVAR(file_list_spec, $1)='@'
   # Tell ltmain to make .lib files, not .a files.
   libext=lib
   # Tell ltmain to make .dll files, not .so files.
   shrext_cmds=.dll
   # FIXME: Setting linknames here is a bad hack.
   _LT_TAGVAR(archive_cmds, $1)='$CC -o $output_objdir/$soname $libobjs $compiler_flags $deplibs -Wl,-
DLL,-IMPLIB:"$tool_output_objdir$libname.dll.lib"~linknames='
   _LT_TAGVAR(archive_expsym_cmds, $1)='if _LT_DLL_DEF_P([$export_symbols]); then
             cp "$export_symbols" "$output_objdir/$soname.def";
             echo "$tool_output_objdir$soname.def" > "$output_objdir/$soname.exp";
           else
	  $SED -e '\''s/^/-link -EXPORT:/'\'' < $export_symbols > $output_objdir/$soname.exp;
	fi~
	$CC -o $tool_output_objdir$soname $libobjs $compiler_flags $deplibs "@$tool_output_objdir$soname.exp"
-Wl,-DLL,-IMPLIB:"$tool_output_objdir$libname.dll.lib"~
	linknames='
   # The linker will not automatically build a static lib if we build a DLL.
   # _LT_TAGVAR(old_archive_from_new_cmds, $1)='true'
   _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=yes
   # Don't use ranlib
   _LT_TAGVAR(old_postinstall_cmds, $1)='chmod 644 $oldlib'
   _LT_TAGVAR(postlink_cmds, $1)='lt_outputfile="@OUTPUT@"~
	lt_tool_outputfile="@TOOL_OUTPUT@"~
	case $lt_outputfile in
	  *.exe|*.EXE) ;;
	  *)
	    lt_outputfile=$lt_outputfile.exe
	    lt_tool_outputfile=$lt_tool_outputfile.exe
	    ;;
	esac~
	func_to_tool_file "$lt_outputfile"~
	if test : != "$MANIFEST_TOOL" && test -f "$lt_outputfile.manifest"; then
	  $MANIFEST_TOOL -manifest "$lt_tool_outputfile.manifest" -outputresource:"$lt_tool_outputfile" || exit 1;
	  $RM "$lt_outputfile.manifest";
	fi'
   ;;
  *)
   # g++
   # _LT_TAGVAR(hardcode_libdir_flag_spec, $1) is actually meaningless,
   # as there is no search path for DLLs.
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
   _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl--export-all-symbols'
   _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
   _LT_TAGVAR(always_export_symbols, $1)=no
   _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=yes

   if $LD --help 2>&1 | $GREP 'auto-import' > /dev/null; then
     _LT_TAGVAR(archive_cmds, $1)='$CC -shared -nostdlib $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags -o $output_objdir/$soname $wl--enable-auto-image-base -Xlinker --out-implib -Xlinker $lib'
     # If the export-symbols file already is a .def file, use it as
     # is; otherwise, prepend EXPORTS...
     _LT_TAGVAR(archive_expsym_cmds, $1)='if _LT_DLL_DEF_P([$export_symbols]); then
	  cp $export_symbols $output_objdir/$soname.def;
	else
	  echo EXPORTS > $output_objdir/$soname.def;
	  cat $export_symbols >> $output_objdir/$soname.def;
	fi~
	$CC -shared -nostdlib $output_objdir/$soname.def $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags -o $output_objdir/$soname $wl--enable-auto-image-base -Xlinker --out-implib -Xlinker $lib'
```

```
  else
    _LT_TAGVAR(ld_shlibs, $1)=no
  fi
  ;;
 esac
 ;;
    darwin* | rhapsody*)
      _LT_DARWIN_LINKER_FEATURES($1)
 ;;

    os2*)
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-L$libdir'
 _LT_TAGVAR(hardcode_minus_L, $1)=yes
 _LT_TAGVAR(allow_undefined_flag, $1)=unsupported
 shrext_cmds=.dll
 _LT_TAGVAR(archive_cmds, $1)='$ECHO "LIBRARY ${soname%$shared_ext} INITINSTANCE
TERMINSTANCE" > $output_objdir/$libname.def~
  $ECHO "DESCRIPTION \"$libname\"" >> $output_objdir/$libname.def~
  $ECHO "DATA MULTIPLE NONSHARED" >> $output_objdir/$libname.def~
  $ECHO EXPORTS >> $output_objdir/$libname.def~
  emxexp $libobjs | $SED /"_DLL_InitTerm"/d >> $output_objdir/$libname.def~
  $CC -Zdll -Zcrtdll -o $output_objdir/$soname $libobjs $deplibs $compiler_flags $output_objdir/$libname.def~
  emximp -o $lib $output_objdir/$libname.def'
 _LT_TAGVAR(archive_expsym_cmds, $1)='$ECHO "LIBRARY ${soname%$shared_ext} INITINSTANCE
TERMINSTANCE" > $output_objdir/$libname.def~
  $ECHO "DESCRIPTION \"$libname\"" >> $output_objdir/$libname.def~
  $ECHO "DATA MULTIPLE NONSHARED" >> $output_objdir/$libname.def~
  $ECHO EXPORTS >> $output_objdir/$libname.def~
  prefix_cmds="$SED"~
  if test EXPORTS = "`$SED 1q $export_symbols`"; then
    prefix_cmds="$prefix_cmds -e 1d";
  fi~
  prefix_cmds="$prefix_cmds -e \"s/^\(.*\)$/_\1/g\""~
  cat $export_symbols | $prefix_cmds >> $output_objdir/$libname.def~
  $CC -Zdll -Zcrtdll -o $output_objdir/$soname $libobjs $deplibs $compiler_flags $output_objdir/$libname.def~
  emximp -o $lib $output_objdir/$libname.def'
 _LT_TAGVAR(old_archive_From_new_cmds, $1)='emximp -o $output_objdir/${libname}_dll.a
$output_objdir/$libname.def'
 _LT_TAGVAR(enable_shared_with_static_runtimes, $1)=yes
 ;;

    dgux*)
      case $cc_basename in
        ec++*)
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
        ghcx*)
    # Green Hills C++ Compiler
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
        *)
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
      esac
      ;;

    freebsd2.*)
      # C++ shared libraries reported to be fairly broken before
  # switch to ELF
      _LT_TAGVAR(ld_shlibs, $1)=no
      ;;

    freebsd-elf*)
      _LT_TAGVAR(archive_cmds_need_lc, $1)=no
      ;;

    freebsd* | dragonfly*)
      # FreeBSD 3 and later use GNU C++ and GNU ld with standard ELF
      # conventions
      _LT_TAGVAR(ld_shlibs, $1)=yes
      ;;

    haiku*)
      _LT_TAGVAR(archive_cmds, $1)='$CC -shared $libobjs $deplibs $compiler_flags $wl-soname $wl$soname -
o $lib'
      _LT_TAGVAR(link_all_deplibs, $1)=yes
      ;;

    hpux9*)
      _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl+b $wl$libdir'
      _LT_TAGVAR(hardcode_libdir_separator, $1)=:
      _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'
      _LT_TAGVAR(hardcode_direct, $1)=yes
      _LT_TAGVAR(hardcode_minus_L, $1)=yes # Not in the search PATH,
              # but as the default
              # location of the library.

      case $cc_basename in
        CC*)
          # FIXME: insert proper C++ library support
          _LT_TAGVAR(ld_shlibs, $1)=no
          ;;
        aCC*)
```

```
	    _LT_TAGVAR(archive_cmds, $1)='$RM $output_objdir/$soname~$CC -b $wl+b $wl$install_libdir -o $output_objdir/$soname $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags~test "x$output_objdir/$soname" = "x$lib" || mv -f $output_objdir/$soname $lib'
	    # Commands to make compiler produce verbose output that lists
	    # what "hidden" libraries, object files and flags are used when
	    # linking a shared library.
	    #
	    # There doesn't appear to be a way to prevent this compiler from
	    # explicitly linking system object files so we need to strip them
	    # from the output so that they don't get included in the library
	    # dependencies.
	    output_verbose_link_cmd='templist=`($CC -b $CFLAGS -v conftest.$objext 2>&1) | $EGREP "\-L"`; list= ; for z in $templist; do case $z in conftest.$objext) list="$list $z";; *.$objext);; *) list="$list $z";;esac; done; func_echo_all "$list"'
	    ;;
	  *)
	    if test yes = "$GXX"; then
	      _LT_TAGVAR(archive_cmds, $1)='$RM $output_objdir/$soname~$CC -shared -nostdlib $pic_flag $wl+b $wl$install_libdir -o $output_objdir/$soname $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags~test "x$output_objdir/$soname" = "x$lib" || mv -f $output_objdir/$soname $lib'
	    else
	      # FIXME: insert proper C++ library support
	      _LT_TAGVAR(ld_shlibs, $1)=no
	    fi
	    ;;
	esac
	;;

    hpux10*|hpux11*)
	if test no = "$with_gnu_ld"; then
	  _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl+b $wl$libdir'
	  _LT_TAGVAR(hardcode_libdir_separator, $1)=:

          case $host_cpu in
            hppa*64*|ia64*)
              ;;
            *)
	      _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'
              ;;
          esac
	fi
	case $host_cpu in
	  hppa*64*|ia64*)
	    _LT_TAGVAR(hardcode_direct, $1)=no
	    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
	    ;;
	  *)
	    _LT_TAGVAR(hardcode_direct, $1)=yes
```

```
          _LT_TAGVAR(hardcode_direct_absolute, $1)=yes
          _LT_TAGVAR(hardcode_minus_L, $1)=yes # Not in the search PATH,
             # but as the default
             # location of the library.
          ;;
      esac

      case $cc_basename in
        CC*)
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
        aCC*)
    case $host_cpu in
      hppa*64*)
        _LT_TAGVAR(archive_cmds, $1)='$CC -b $wl+h $wl$soname -o $lib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags'
        ;;
      ia64*)
        _LT_TAGVAR(archive_cmds, $1)='$CC -b $wl+h $wl$soname $wl+nodefaultrpath -o $lib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags'
        ;;
      *)
        _LT_TAGVAR(archive_cmds, $1)='$CC -b $wl+h $wl$soname $wl+b $wl$install_libdir -o $lib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags'
        ;;
    esac
    # Commands to make compiler produce verbose output that lists
    # what "hidden" libraries, object files and flags are used when
    # linking a shared library.
    #
    # There doesn't appear to be a way to prevent this compiler from
    # explicitly linking system object files so we need to strip them
    # from the output so that they don't get included in the library
    # dependencies.
    output_verbose_link_cmd='templist=`($CC -b $CFLAGS -v conftest.$objext 2>&1) | $GREP "\-L"`; list= ; for z in $templist; do case $z in conftest.$objext) list="$list $z";; *.$objext);; *) list="$list $z";;esac; done; func_echo_all "$list"'
    ;;
        *)
    if test yes = "$GXX"; then
      if test no = "$with_gnu_ld"; then
        case $host_cpu in
          hppa*64*)
            _LT_TAGVAR(archive_cmds, $1)='$CC -shared -nostdlib -fPIC $wl+h $wl$soname -o $lib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags'
            ;;
          ia64*)
```

```
            _LT_TAGVAR(archive_cmds, $1)='$CC -shared -nostdlib $pic_flag $wl+h $wl$soname $wl+nodefaultrpath -o $lib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags'
            ;;
          *)
            _LT_TAGVAR(archive_cmds, $1)='$CC -shared -nostdlib $pic_flag $wl+h $wl$soname $wl+b $wl$install_libdir -o $lib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags'
            ;;
        esac
      fi
    else
      # FIXME: insert proper C++ library support
      _LT_TAGVAR(ld_shlibs, $1)=no
    fi
    ;;
      esac
      ;;

    interix[[3-9]]*)
 _LT_TAGVAR(hardcode_direct, $1)=no
 _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
 _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath,$libdir'
 _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'
 # Hack: On Interix 3.x, we cannot compile PIC because of a broken gcc.
 # Instead, shared libraries are loaded at an image base (0x10000000 by
 # default) and relocated if they conflict, which is a slow very memory
 # consuming and fragmenting process.  To avoid this, we pick a random,
 # 256 KiB-aligned image base between 0x50000000 and 0x6FFC0000 at link
 # time.  Moving up from 0x10000000 also allows more sbrk(2) space.
 _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-h,$soname $wl--image-base,`expr ${RANDOM-$$} % 4096 / 2 \* 262144 + 1342177280` -o $lib'
 _LT_TAGVAR(archive_expsym_cmds, $1)='sed "s|^|_|" $export_symbols >$output_objdir/$soname.expsym~$CC -shared $pic_flag $libobjs $deplibs $compiler_flags $wl-h,$soname $wl--retain-symbols-file,$output_objdir/$soname.expsym $wl--image-base,`expr ${RANDOM-$$} % 4096 / 2 \* 262144 + 1342177280` -o $lib'
 ;;
    irix5* | irix6*)
      case $cc_basename in
        CC*)
    # SGI C++
    _LT_TAGVAR(archive_cmds, $1)='$CC -shared -all -multigot $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags -soname $soname `test -n "$verstring" && func_echo_all "-set_version $verstring"` -update_registry $output_objdir/so_locations -o $lib'

    # Archives containing C++ object files must be created using
    # "CC -ar", where "CC" is the IRIX C++ compiler.  This is
    # necessary to make sure instantiated templates are included
    # in the archive.
    _LT_TAGVAR(old_archive_cmds, $1)='$CC -ar -WR,-u -o $oldlib $oldobjs'
```

```
  ;;
    *)
  if test yes = "$GXX"; then
   if test no = "$with_gnu_ld"; then
    _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag -nostdlib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags $wl-soname $wl$soname `test -n "$verstring" && func_echo_all "$wl-set_version $wl$verstring"` $wl-update_registry $wl$output_objdir/so_locations -o $lib'
   else
    _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag -nostdlib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags $wl-soname $wl$soname `test -n "$verstring" && func_echo_all "$wl-set_version $wl$verstring"` -o $lib'
   fi
  fi
  _LT_TAGVAR(link_all_deplibs, $1)=yes
  ;;
   esac
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
   _LT_TAGVAR(hardcode_libdir_separator, $1)=:
   _LT_TAGVAR(inherit_rpath, $1)=yes
   ;;

  linux* | k*bsd*-gnu | kopensolaris*-gnu | gnu*)
   case $cc_basename in
    KCC*)
  # Kuck and Associates, Inc. (KAI) C++ Compiler

  # KCC will only create a shared library if the output file
  # ends with ".so" (or ".sl" for HP-UX), so rename the library
  # to its proper name (with version) after linking.
  _LT_TAGVAR(archive_cmds, $1)='tempext=`echo $shared_ext | $SED -e '\''s/\([[^()0-9A-Za-z{}]]\)/\\\\\1/g'\''`; templib=`echo $lib | $SED -e "s/\$tempext\..*/.so/"`; $CC $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags --soname $soname -o \$templib; mv -f \$templib $lib'
  _LT_TAGVAR(archive_expsym_cmds, $1)='tempext=`echo $shared_ext | $SED -e '\''s/\([[^()0-9A-Za-z{}]]\)/\\\\\1/g'\''`; templib=`echo $lib | $SED -e "s/\$tempext\..*/.so/"`; $CC $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags --soname $soname -o \$templib $wl-retain-symbols-file,$export_symbols; mv -f \$templib $lib'
  # Commands to make compiler produce verbose output that lists
  # what "hidden" libraries, object files and flags are used when
  # linking a shared library.
  #
  # There doesn't appear to be a way to prevent this compiler from
  # explicitly linking system object files so we need to strip them
  # from the output so that they don't get included in the library
  # dependencies.
  output_verbose_link_cmd='templist=`$CC $CFLAGS -v conftest.$objext -o libconftest$shared_ext 2>&1 | $GREP "ld"`; rm -f libconftest$shared_ext; list= ; for z in $templist; do case $z in conftest.$objext) list="$list $z";; *.$objext);; *) list="$list $z";;esac; done; func_echo_all "$list"'
```

```
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath,$libdir'
     _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl--export-dynamic'

     # Archives containing C++ object files must be created using
     # "CC -Bstatic", where "CC" is the KAI C++ compiler.
     _LT_TAGVAR(old_archive_cmds, $1)='$CC -Bstatic -o $oldlib $oldobjs'
     ;;
   icpc* | ecpc* )
     # Intel C++
     with_gnu_ld=yes
     # version 8.0 and above of icpc choke on multiply defined symbols
     # if we add $predep_objects and $postdep_objects, however 7.1 and
     # earlier do not add the objects themselves.
     case `$CC -V 2>&1` in
       *"Version 7."*)
         _LT_TAGVAR(archive_cmds, $1)='$CC -shared $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags $wl-soname $wl$soname -o $lib'
  _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags $wl-soname $wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'
  ;;
       *)  # Version 8.0 or newer
         tmp_idyn=
         case $host_cpu in
    ia64*) tmp_idyn=' -i_dynamic';;
  esac
         _LT_TAGVAR(archive_cmds, $1)='$CC -shared'"$tmp_idyn"' $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname -o $lib'
  _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared'"$tmp_idyn"' $libobjs $deplibs $compiler_flags $wl-
soname $wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'
  ;;
     esac
     _LT_TAGVAR(archive_cmds_need_lc, $1)=no
     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath,$libdir'
     _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl--export-dynamic'
     _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive$convenience $wl--no-whole-archive'
     ;;
         pgCC* | pgcpp*)
           # Portland Group C++ compiler
     case `$CC -V` in
     *pgCC\ [[1-5]].* | *pgcpp\ [[1-5]].*)
       _LT_TAGVAR(prelink_cmds, $1)='tpldir=Template.dir~
              rm -rf $tpldir~
              $CC --prelink_objects --instantiation_dir $tpldir $objs $libobjs $compile_deplibs~
              compile_command="$compile_command `find $tpldir -name \*.o | sort | $NL2SP`"'
       _LT_TAGVAR(old_archive_cmds, $1)='tpldir=Template.dir~
               rm -rf $tpldir~
               $CC --prelink_objects --instantiation_dir $tpldir $oldobjs$old_deplibs~
               $AR $AR_FLAGS $oldlib$oldobjs$old_deplibs `find $tpldir -name \*.o | sort | $NL2SP`~
```

```
          $RANLIB $oldlib'
      _LT_TAGVAR(archive_cmds, $1)='tpldir=Template.dir~
          rm -rf $tpldir~
          $CC --prelink_objects --instantiation_dir $tpldir $predep_objects $libobjs $deplibs $convenience
$postdep_objects~
          $CC -shared $pic_flag $predep_objects $libobjs $deplibs `find $tpldir -name \*.o | sort | $NL2SP`
$postdep_objects $compiler_flags $wl-soname $wl$soname -o $lib'
      _LT_TAGVAR(archive_expsym_cmds, $1)='tpldir=Template.dir~
          rm -rf $tpldir~
          $CC --prelink_objects --instantiation_dir $tpldir $predep_objects $libobjs $deplibs $convenience
$postdep_objects~
          $CC -shared $pic_flag $predep_objects $libobjs $deplibs `find $tpldir -name \*.o | sort | $NL2SP`
$postdep_objects $compiler_flags $wl-soname $wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'
      ;;
    *) # Version 6 and above use weak symbols
      _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags $wl-soname $wl$soname -o $lib'
      _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $pic_flag $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags $wl-soname $wl$soname $wl-retain-symbols-file $wl$export_symbols -o $lib'
      ;;
    esac

    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl--rpath $wl$libdir'
    _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl--export-dynamic'
    _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive`for conv in $convenience\"\"; do test  -n
\"$conv\" && new_convenience=\"$new_convenience,$conv\"; done; func_echo_all \"$new_convenience\"` $wl--
no-whole-archive'
        ;;
  cxx*)
    # Compaq C++
    _LT_TAGVAR(archive_cmds, $1)='$CC -shared $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags $wl-soname $wl$soname -o $lib'
    _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags $wl-soname $wl$soname  -o $lib $wl-retain-symbols-file $wl$export_symbols'

    runpath_var=LD_RUN_PATH
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-rpath $libdir'
    _LT_TAGVAR(hardcode_libdir_separator, $1)=:

    # Commands to make compiler produce verbose output that lists
    # what "hidden" libraries, object files and flags are used when
    # linking a shared library.
    #
    # There doesn't appear to be a way to prevent this compiler from
    # explicitly linking system object files so we need to strip them
    # from the output so that they don't get included in the library
    # dependencies.
    output_verbose_link_cmd='templist=`$CC -shared $CFLAGS -v conftest.$objext 2>&1 | $GREP "ld"`;
```

```
templist=`func_echo_all "$templist" | $SED "s/\(^.*ld.*\)\( .*ld .*$\)/\1/"`; list= ; for z in $templist; do case $z in
conftest.$objext) list="$list $z";; *.$objext);; *) list="$list $z";;esac; done; func_echo_all "X$list" | $Xsed'
    ;;
  xl* | mpixl* | bgxl*)
    # IBM XL 8.0 on PPC, with GNU ld
    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
    _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl--export-dynamic'
    _LT_TAGVAR(archive_cmds, $1)='$CC -qmkshrobj $libobjs $deplibs $compiler_flags $wl-soname
$wl$soname -o $lib'
    if test yes = "$supports_anon_versioning"; then
      _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $output_objdir/$libname.ver~
            cat $export_symbols | sed -e "s/\(.*\)/\1;/" >> $output_objdir/$libname.ver~
            echo "local: *; };" >> $output_objdir/$libname.ver~
            $CC -qmkshrobj $libobjs $deplibs $compiler_flags $wl-soname $wl$soname $wl-version-script
$wl$output_objdir/$libname.ver -o $lib'
    fi
    ;;
  *)
    case `$CC -V 2>&1 | sed 5q` in
    *Sun\ C*)
      # Sun C++ 5.9
      _LT_TAGVAR(no_undefined_flag, $1)=' -zdefs'
      _LT_TAGVAR(archive_cmds, $1)='$CC -G$allow_undefined_flag -h$soname -o $lib $predep_objects $libobjs
$deplibs $postdep_objects $compiler_flags'
      _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -G$allow_undefined_flag -h$soname -o $lib $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags $wl-retain-symbols-file $wl$export_symbols'
      _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-R$libdir'
      _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl--whole-archive`new_convenience=; for conv in
$convenience\"\"; do test -z \"$conv\" || new_convenience=\"$new_convenience,$conv\"; done; func_echo_all
\"$new_convenience\"` $wl--no-whole-archive'
      _LT_TAGVAR(compiler_needs_object, $1)=yes

      # Not sure whether something based on
      # $CC $CFLAGS -v conftest.$objext -o libconftest$shared_ext 2>&1
      # would be better.
      output_verbose_link_cmd='func_echo_all'

      # Archives containing C++ object files must be created using
      # "CC -xar", where "CC" is the Sun C++ compiler.  This is
      # necessary to make sure instantiated templates are included
      # in the archive.
      _LT_TAGVAR(old_archive_cmds, $1)='$CC -xar -o $oldlib $oldobjs'
      ;;
    esac
    ;;
 esac
 ;;
```

```
    lynxos*)
      # FIXME: insert proper C++ library support
 _LT_TAGVAR(ld_shlibs, $1)=no
 ;;

    m88k*)
      # FIXME: insert proper C++ library support
      _LT_TAGVAR(ld_shlibs, $1)=no
 ;;

    mvs*)
      case $cc_basename in
        cxx*)
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
   *)
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
 esac
 ;;

    netbsd*)
      if echo __ELF__ | $CC -E - | $GREP __ELF__ >/dev/null; then
   _LT_TAGVAR(archive_cmds, $1)='$LD -Bshareable  -o $lib $predep_objects $libobjs $deplibs $postdep_objects
$linker_flags'
   wlarc=
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-R$libdir'
   _LT_TAGVAR(hardcode_direct, $1)=yes
   _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
 fi
 # Workaround some broken pre-1.5 toolchains
 output_verbose_link_cmd='$CC -shared $CFLAGS -v conftest.$objext 2>&1 | $GREP conftest.$objext | $SED -e
"s:-lgcc -lc -lgcc::"'
 ;;

    *nto* | *qnx*)
      _LT_TAGVAR(ld_shlibs, $1)=yes
 ;;

    openbsd* | bitrig*)
 if test -f /usr/libexec/ld.so; then
   _LT_TAGVAR(hardcode_direct, $1)=yes
   _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
   _LT_TAGVAR(hardcode_direct_absolute, $1)=yes
   _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags -o $lib'
```

```
   _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath,$libdir'
   if test -z "`echo __ELF__ | $CC -E - | grep __ELF__`"; then
     _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $pic_flag $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags $wl-retain-symbols-file,$export_symbols -o $lib'
     _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-E'
     _LT_TAGVAR(whole_archive_flag_spec, $1)=$wlarc'--whole-archive$convenience '$wlarc'--no-whole-archive'
   fi
   output_verbose_link_cmd=func_echo_all
 else
   _LT_TAGVAR(ld_shlibs, $1)=no
 fi
 ;;

     osf3* | osf4* | osf5*)
       case $cc_basename in
         KCC*)
     # Kuck and Associates, Inc. (KAI) C++ Compiler

     # KCC will only create a shared library if the output file
     # ends with ".so" (or ".sl" for HP-UX), so rename the library
     # to its proper name (with version) after linking.
     _LT_TAGVAR(archive_cmds, $1)='tempext=`echo $shared_ext | $SED -e '\''s/\([[^()0-9A-Za-z{}]]\)/\\\\\1/g'\''`; templib=`echo "$lib" | $SED -e "s/\$tempext\..*/.so/"`; $CC $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags --soname $soname -o \$templib; mv -f \$templib $lib'

     _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath,$libdir'
     _LT_TAGVAR(hardcode_libdir_separator, $1)=:

     # Archives containing C++ object files must be created using
     # the KAI C++ compiler.
     case $host in
       osf3*) _LT_TAGVAR(old_archive_cmds, $1)='$CC -Bstatic -o $oldlib $oldobjs' ;;
       *) _LT_TAGVAR(old_archive_cmds, $1)='$CC -o $oldlib $oldobjs' ;;
     esac
     ;;
         RCC*)
     # Rational C++ 2.4.1
     # FIXME: insert proper C++ library support
     _LT_TAGVAR(ld_shlibs, $1)=no
     ;;
         cxx*)
     case $host in
       osf3*)
         _LT_TAGVAR(allow_undefined_flag, $1)=' $wl-expect_unresolved $wl\*'
         _LT_TAGVAR(archive_cmds, $1)='$CC -shared$allow_undefined_flag $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags $wl-soname $soname `test -n "$verstring" && func_echo_all "$wl-set_version $verstring"` -update_registry $output_objdir/so_locations -o $lib'
         _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
```

```
  ;;
      *)
        _LT_TAGVAR(allow_undefined_flag, $1)=' -expect_unresolved \*'
        _LT_TAGVAR(archive_cmds, $1)='$CC -shared$allow_undefined_flag $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags -msym -soname $soname `test -n "$verstring" && func_echo_all "-set_version
$verstring"` -update_registry $output_objdir/so_locations -o $lib'
        _LT_TAGVAR(archive_expsym_cmds, $1)='for i in `cat $export_symbols`; do printf "%s %s\\n" -
exported_symbol "\$i" >> $lib.exp; done~
                echo "-hidden">> $lib.exp~
                $CC -shared$allow_undefined_flag $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags
-msym -soname $soname $wl-input $wl$lib.exp  `test -n "$verstring" && $ECHO "-set_version $verstring"` -
update_registry $output_objdir/so_locations -o $lib~
                $RM $lib.exp'
        _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-rpath $libdir'
  ;;
    esac

    _LT_TAGVAR(hardcode_libdir_separator, $1)=:

    # Commands to make compiler produce verbose output that lists
    # what "hidden" libraries, object files and flags are used when
    # linking a shared library.
    #
    # There doesn't appear to be a way to prevent this compiler from
    # explicitly linking system object files so we need to strip them
    # from the output so that they don't get included in the library
    # dependencies.
    output_verbose_link_cmd='templist=`$CC -shared $CFLAGS -v conftest.$objext 2>&1 | $GREP "ld" | $GREP -v
"ld:"`; templist=`func_echo_all "$templist" | $SED "s/\(^.*ld.*\)\( .*ld.*$\)/\1/"`; list= ; for z in $templist; do case $z
in conftest.$objext) list="$list $z";; *.$objext);; *) list="$list $z";;esac; done; func_echo_all "$list"'
    ;;
   *)
    if test yes,no = "$GXX,$with_gnu_ld"; then
      _LT_TAGVAR(allow_undefined_flag, $1)=' $wl-expect_unresolved $wl\*'
      case $host in
        osf3*)
          _LT_TAGVAR(archive_cmds, $1)='$CC -shared -nostdlib $allow_undefined_flag $predep_objects $libobjs
$deplibs $postdep_objects $compiler_flags $wl-soname $wl$soname `test -n "$verstring" && func_echo_all "$wl-
set_version $wl$verstring"` $wl-update_registry $wl$output_objdir/so_locations -o $lib'
   ;;
        *)
          _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag -nostdlib $allow_undefined_flag $predep_objects
$libobjs $deplibs $postdep_objects $compiler_flags $wl-msym $wl-soname $wl$soname `test -n "$verstring" &&
func_echo_all "$wl-set_version $wl$verstring"` $wl-update_registry $wl$output_objdir/so_locations -o $lib'
   ;;
      esac

      _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-rpath $wl$libdir'
      _LT_TAGVAR(hardcode_libdir_separator, $1)=:

      # Commands to make compiler produce verbose output that lists
      # what "hidden" libraries, object files and flags are used when
      # linking a shared library.
      output_verbose_link_cmd='$CC -shared $CFLAGS -v conftest.$objext 2>&1 | $GREP -v "^Configured with:" |
$GREP "\-L"'

    else
      # FIXME: insert proper C++ library support
      _LT_TAGVAR(ld_shlibs, $1)=no
    fi
    ;;
      esac
      ;;

    psos*)
      # FIXME: insert proper C++ library support
      _LT_TAGVAR(ld_shlibs, $1)=no
      ;;

    sunos4*)
      case $cc_basename in
        CC*)
    # Sun C++ 4.x
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
        lcc*)
    # Lucid
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
        *)
    # FIXME: insert proper C++ library support
    _LT_TAGVAR(ld_shlibs, $1)=no
    ;;
      esac
      ;;

    solaris*)
      case $cc_basename in
        CC* | sunCC*)
    # Sun C++ 4.2, 5.x and Centerline C++
          _LT_TAGVAR(archive_cmds_need_lc,$1)=yes
    _LT_TAGVAR(no_undefined_flag, $1)=' -zdefs'
    _LT_TAGVAR(archive_cmds, $1)='$CC -G$allow_undefined_flag -h$soname -o $lib $predep_objects $libobjs
$deplibs $postdep_objects $compiler_flags'
```

```
    _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $lib.exp~cat $export_symbols | $SED -e
"s/\(.*\)/\1;/" >> $lib.exp~echo "local: *; };" >> $lib.exp~
          $CC -G$allow_undefined_flag $wl-M $wl$lib.exp -h$soname -o $lib $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags~$RM $lib.exp'

    _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='-R$libdir'
    _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
    case $host_os in
      solaris2.[[0-5]] | solaris2.[[0-5]].*) ;;
      *)
  # The compiler driver will combine and reorder linker options,
  # but understands '-z linker_flag'.
        # Supported since Solaris 2.6 (maybe 2.5.1?)
  _LT_TAGVAR(whole_archive_flag_spec, $1)='-z allextract$convenience -z defaultextract'
        ;;
    esac
    _LT_TAGVAR(link_all_deplibs, $1)=yes

    output_verbose_link_cmd='func_echo_all'

    # Archives containing C++ object files must be created using
    # "CC -xar", where "CC" is the Sun C++ compiler.  This is
    # necessary to make sure instantiated templates are included
    # in the archive.
    _LT_TAGVAR(old_archive_cmds, $1)='$CC -xar -o $oldlib $oldobjs'
    ;;
        gcx*)
    # Green Hills C++ Compiler
    _LT_TAGVAR(archive_cmds, $1)='$CC -shared $predep_objects $libobjs $deplibs $postdep_objects
$compiler_flags $wl-h $wl$soname -o $lib'

    # The C++ compiler must be used to create the archive.
    _LT_TAGVAR(old_archive_cmds, $1)='$CC $LDFLAGS -archive -o $oldlib $oldobjs'
    ;;
        *)
    # GNU C++ compiler with Solaris linker
    if test yes,no = "$GXX,$with_gnu_ld"; then
      _LT_TAGVAR(no_undefined_flag, $1)=' $wl-z ${wl}defs'
      if $CC --version | $GREP -v '^2\.7' > /dev/null; then
        _LT_TAGVAR(archive_cmds, $1)='$CC -shared $pic_flag -nostdlib $predep_objects $libobjs $deplibs
$postdep_objects $compiler_flags $wl-h $wl$soname -o $lib'
        _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $lib.exp~cat $export_symbols | $SED -e
"s/\(.*\)/\1;/" >> $lib.exp~echo "local: *; };" >> $lib.exp~
              $CC -shared $pic_flag -nostdlib $wl-M $wl$lib.exp $wl-h $wl$soname -o $lib $predep_objects $libobjs
$deplibs $postdep_objects $compiler_flags~$RM $lib.exp'

        # Commands to make compiler produce verbose output that lists
        # what "hidden" libraries, object files and flags are used when
```

```
         # linking a shared library.
         output_verbose_link_cmd='$CC -shared $CFLAGS -v conftest.$objext 2>&1 | $GREP -v "^Configured with:" | $GREP "\-L"'
       else
         # g++ 2.7 appears to require '-G' NOT '-shared' on this
         # platform.
         _LT_TAGVAR(archive_cmds, $1)='$CC -G -nostdlib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags $wl-h $wl$soname -o $lib'
         _LT_TAGVAR(archive_expsym_cmds, $1)='echo "{ global:" > $lib.exp~cat $export_symbols | $SED -e "s/\(.*\)/\1;/" >> $lib.exp~echo "local: *; };" >> $lib.exp~
                 $CC -G -nostdlib $wl-M $wl$lib.exp $wl-h $wl$soname -o $lib $predep_objects $libobjs $deplibs $postdep_objects $compiler_flags~$RM $lib.exp'

         # Commands to make compiler produce verbose output that lists
         # what "hidden" libraries, object files and flags are used when
         # linking a shared library.
         output_verbose_link_cmd='$CC -G $CFLAGS -v conftest.$objext 2>&1 | $GREP -v "^Configured with:" | $GREP "\-L"'
       fi

       _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-R $wl$libdir'
       case $host_os in
  solaris2.[[0-5]] | solaris2.[[0-5]].*) ;;
  *)
    _LT_TAGVAR(whole_archive_flag_spec, $1)='$wl-z ${wl}allextract$convenience $wl-z ${wl}defaultextract'
    ;;
       esac
     fi
     ;;
       esac
       ;;

   sysv4*uw2* | sysv5OpenUNIX* | sysv5UnixWare7.[[01]].[[10]]* | unixware7* | sco3.2v5.0.[[024]]*)
     _LT_TAGVAR(no_undefined_flag, $1)='$wl-z,text'
     _LT_TAGVAR(archive_cmds_need_lc, $1)=no
     _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
     runpath_var='LD_RUN_PATH'

     case $cc_basename in
       CC*)
   _LT_TAGVAR(archive_cmds, $1)='$CC -G $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
   _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -G $wl-Bexport:$export_symbols $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
   ;;
  *)
   _LT_TAGVAR(archive_cmds, $1)='$CC -shared $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
   _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $wl-Bexport:$export_symbols $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
   ;;
     esac
     ;;

     sysv5* | sco3.2v5* | sco5v6*)
  # Note: We CANNOT use -z defs as we might desire, because we do not
  # link with -lc, and that would cause any symbols used from libc to
  # always be unresolved, which means just about no library would
  # ever link correctly.  If we're not using GNU ld we use -z text
  # though, which does catch some bad symbols but isn't as heavy-handed
  # as -z defs.
  _LT_TAGVAR(no_undefined_flag, $1)='$wl-z,text'
  _LT_TAGVAR(allow_undefined_flag, $1)='$wl-z,nodefs'
  _LT_TAGVAR(archive_cmds_need_lc, $1)=no
  _LT_TAGVAR(hardcode_shlibpath_var, $1)=no
  _LT_TAGVAR(hardcode_libdir_flag_spec, $1)='$wl-R,$libdir'
  _LT_TAGVAR(hardcode_libdir_separator, $1)=':'
  _LT_TAGVAR(link_all_deplibs, $1)=yes
  _LT_TAGVAR(export_dynamic_flag_spec, $1)='$wl-Bexport'
  runpath_var='LD_RUN_PATH'

  case $cc_basename in
         CC*)
     _LT_TAGVAR(archive_cmds, $1)='$CC -G $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
     _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -G $wl-Bexport:$export_symbols $wl-h,$soname -o $lib
 $libobjs $deplibs $compiler_flags'
     _LT_TAGVAR(old_archive_cmds, $1)='$CC -Tprelink_objects $oldobjs~
          '"$_LT_TAGVAR(old_archive_cmds, $1)"
     _LT_TAGVAR(reload_cmds, $1)='$CC -Tprelink_objects $reload_objs~
          '"$_LT_TAGVAR(reload_cmds, $1)"
     ;;
   *)
     _LT_TAGVAR(archive_cmds, $1)='$CC -shared $wl-h,$soname -o $lib $libobjs $deplibs $compiler_flags'
     _LT_TAGVAR(archive_expsym_cmds, $1)='$CC -shared $wl-Bexport:$export_symbols $wl-h,$soname -o $lib
 $libobjs $deplibs $compiler_flags'
     ;;
  esac
     ;;

     tandem*)
       case $cc_basename in
         NCC*)
     # NonStop-UX NCC 3.20
     # FIXME: insert proper C++ library support
     _LT_TAGVAR(ld_shlibs, $1)=no
     ;;
         *)
     # FIXME: insert proper C++ library support
```

```
	    _LT_TAGVAR(ld_shlibs, $1)=no
	    ;;
	  esac
	  ;;

	vxworks*)
	  # FIXME: insert proper C++ library support
	  _LT_TAGVAR(ld_shlibs, $1)=no
	  ;;

	*)
	  # FIXME: insert proper C++ library support
	  _LT_TAGVAR(ld_shlibs, $1)=no
	  ;;
    esac

    AC_MSG_RESULT([$_LT_TAGVAR(ld_shlibs, $1)])
    test no = "$_LT_TAGVAR(ld_shlibs, $1)" && can_build_shared=no

    _LT_TAGVAR(GCC, $1)=$GXX
    _LT_TAGVAR(LD, $1)=$LD

    ## CAVEAT EMPTOR:
    ## There is no encapsulation within the following macros, do not change
    ## the running order or otherwise move them around unless you know exactly
    ## what you are doing...
    _LT_SYS_HIDDEN_LIBDEPS($1)
    _LT_COMPILER_PIC($1)
    _LT_COMPILER_C_O($1)
    _LT_COMPILER_FILE_LOCKS($1)
    _LT_LINKER_SHLIBS($1)
    _LT_SYS_DYNAMIC_LINKER($1)
    _LT_LINKER_HARDCODE_LIBPATH($1)

    _LT_CONFIG($1)
  fi # test -n "$compiler"

  CC=$lt_save_CC
  CFLAGS=$lt_save_CFLAGS
  LDCXX=$LD
  LD=$lt_save_LD
  GCC=$lt_save_GCC
  with_gnu_ld=$lt_save_with_gnu_ld
  lt_cv_path_LDCXX=$lt_cv_path_LD
  lt_cv_path_LD=$lt_save_path_LD
  lt_cv_prog_gnu_ldcxx=$lt_cv_prog_gnu_ld
  lt_cv_prog_gnu_ld=$lt_save_with_gnu_ld
fi # test yes != "$_lt_caught_CXX_error"
```

```
AC_LANG_POP
])# _LT_LANG_CXX_CONFIG


# _LT_FUNC_STRIPNAME_CNF
# ----------------------
# func_stripname_cnf prefix suffix name
# strip PREFIX and SUFFIX off of NAME.
# PREFIX and SUFFIX must not contain globbing or regex special
# characters, hashes, percent signs, but SUFFIX may contain a leading
# dot (in which case that matches only a dot).
#
# This function is identical to the (non-XSI) version of func_stripname,
# except this one can be used by m4 code that may be executed by configure,
# rather than the libtool script.
m4_defun([_LT_FUNC_STRIPNAME_CNF],[dnl
AC_REQUIRE([_LT_DECL_SED])
AC_REQUIRE([_LT_PROG_ECHO_BACKSLASH])
func_stripname_cnf ()
{
 case @S|@2 in
 .*) func_stripname_result=`$ECHO "@S|@3" | $SED "s%^@S|@1%%; s%\\\\@S|@2\$%%"`;;
 *)  func_stripname_result=`$ECHO "@S|@3" | $SED "s%^@S|@1%%; s%@S|@2\$%%"`;;
 esac
} # func_stripname_cnf
])# _LT_FUNC_STRIPNAME_CNF


# _LT_SYS_HIDDEN_LIBDEPS([TAGNAME])
# --------------------------------
# Figure out "hidden" library dependencies from verbose
# compiler output when linking a shared library.
# Parse the compiler output and extract the necessary
# objects, libraries and library flags.
m4_defun([_LT_SYS_HIDDEN_LIBDEPS],
[m4_require([_LT_FILEUTILS_DEFAULTS])dnl
AC_REQUIRE([_LT_FUNC_STRIPNAME_CNF])dnl
# Dependencies to place before and after the object being linked:
_LT_TAGVAR(predep_objects, $1)=
_LT_TAGVAR(postdep_objects, $1)=
_LT_TAGVAR(predeps, $1)=
_LT_TAGVAR(postdeps, $1)=
_LT_TAGVAR(compiler_lib_search_path, $1)=

dnl we can't use the lt_simple_compile_test_code here,
dnl because it contains code intended for an executable,
dnl not a library.  It's possible we should let each
dnl tag define a new lt_????_link_test_code variable,
dnl but it's only used here...
m4_if([$1], [], [cat > conftest.$ac_ext <<_LT_EOF
int a;
void foo (void) { a = 0; }
_LT_EOF
], [$1], [CXX], [cat > conftest.$ac_ext <<_LT_EOF
class Foo
{
public:
 Foo (void) { a = 0; }
private:
 int a;
};
_LT_EOF
], [$1], [F77], [cat > conftest.$ac_ext <<_LT_EOF
     subroutine foo
     implicit none
     integer*4 a
     a=0
     return
     end
_LT_EOF
], [$1], [FC], [cat > conftest.$ac_ext <<_LT_EOF
     subroutine foo
     implicit none
     integer a
     a=0
     return
     end
_LT_EOF
], [$1], [GCJ], [cat > conftest.$ac_ext <<_LT_EOF
public class foo {
 private int a;
 public void bar (void) {
   a = 0;
 }
};
_LT_EOF
], [$1], [GO], [cat > conftest.$ac_ext <<_LT_EOF
package foo
func foo() {
}
_LT_EOF
])

_lt_libdeps_save_CFLAGS=$CFLAGS
case "$CC $CFLAGS " in #(
*\ -flto*\ *) CFLAGS="$CFLAGS -fno-lto" ;;
*\ -fwhopr*\ *) CFLAGS="$CFLAGS -fno-whopr" ;;
*\ -fuse-linker-plugin*\ *) CFLAGS="$CFLAGS -fno-use-linker-plugin" ;;
esac

dnl Parse the compiler output and extract the necessary
dnl objects, libraries and library flags.
if AC_TRY_EVAL(ac_compile); then
 # Parse the compiler output and extract the necessary
 # objects, libraries and library flags.

 # Sentinel used to keep track of whether or not we are before
 # the conftest object file.
 pre_test_object_deps_done=no

 for p in `eval "$output_verbose_link_cmd"`; do
   case $prev$p in

   -L* | -R* | -l*)
      # Some compilers place space between "-{L,R}" and the path.
      # Remove the space.
      if test x-L = "$p" ||
         test x-R = "$p"; then
  prev=$p
  continue
      fi

      # Expand the sysroot to ease extracting the directories later.
      if test -z "$prev"; then
        case $p in
        -L*) func_stripname_cnf '-L' '' "$p"; prev=-L; p=$func_stripname_result ;;
        -R*) func_stripname_cnf '-R' '' "$p"; prev=-R; p=$func_stripname_result ;;
        -l*) func_stripname_cnf '-l' '' "$p"; prev=-l; p=$func_stripname_result ;;
        esac
      fi
      case $p in
      =*) func_stripname_cnf '=' '' "$p"; p=$lt_sysroot$func_stripname_result ;;
      esac
      if test no = "$pre_test_object_deps_done"; then
  case $prev in
  -L | -R)
    # Internal compiler library paths should come after those
    # provided the user.  The postdeps already come after the
    # user supplied libs so there is no need to process them.
    if test -z "$_LT_TAGVAR(compiler_lib_search_path, $1)"; then
      _LT_TAGVAR(compiler_lib_search_path, $1)=$prev$p
    else
      _LT_TAGVAR(compiler_lib_search_path, $1)="${_LT_TAGVAR(compiler_lib_search_path, $1)} $prev$p"
	  fi
	  ;;
	# The "-l" case would never come before the object being
	# linked, so don't bother handling this case.
	esac
	    else
	if test -z "$_LT_TAGVAR(postdeps, $1)"; then
	  _LT_TAGVAR(postdeps, $1)=$prev$p
	else
	  _LT_TAGVAR(postdeps, $1)="${_LT_TAGVAR(postdeps, $1)} $prev$p"
	fi
	    fi
	    prev=
	    ;;

	 *.lto.$objext) ;; # Ignore GCC LTO objects
	 *.$objext)
	    # This assumes that the test object file only shows up
	    # once in the compiler output.
	    if test "$p" = "conftest.$objext"; then
	pre_test_object_deps_done=yes
	continue
	    fi

	    if test no = "$pre_test_object_deps_done"; then
	if test -z "$_LT_TAGVAR(predep_objects, $1)"; then
	  _LT_TAGVAR(predep_objects, $1)=$p
	else
	  _LT_TAGVAR(predep_objects, $1)="$_LT_TAGVAR(predep_objects, $1) $p"
	fi
	    else
	if test -z "$_LT_TAGVAR(postdep_objects, $1)"; then
	  _LT_TAGVAR(postdep_objects, $1)=$p
	else
	  _LT_TAGVAR(postdep_objects, $1)="$_LT_TAGVAR(postdep_objects, $1) $p"
	fi
	    fi
	    ;;

	 *) ;; # Ignore the rest.

	 esac
       done

       # Clean up.
       rm -f a.out a.exe
     else
       echo "libtool.m4: error: problem compiling $1 test program"
```

```
fi

$RM -f confest.$objext
CFLAGS=$_lt_libdeps_save_CFLAGS

# PORTME: override above test on systems where it is broken
m4_if([$1], [CXX],
[case $host_os in
interix[[3-9]]*)
 # Interix 3.5 installs completely hosed .la files for C++, so rather than
 # hack all around it, let's just trust "g++" to DTRT.
 _LT_TAGVAR(predep_objects,$1)=
 _LT_TAGVAR(postdep_objects,$1)=
 _LT_TAGVAR(postdeps,$1)=
 ;;
esac
])

case " $_LT_TAGVAR(postdeps, $1) " in
*" -lc "*) _LT_TAGVAR(archive_cmds_need_lc, $1)=no ;;
esac
_LT_TAGVAR(compiler_lib_search_dirs, $1)=
if test -n "${_LT_TAGVAR(compiler_lib_search_path, $1)}"; then
_LT_TAGVAR(compiler_lib_search_dirs, $1)=`echo " ${_LT_TAGVAR(compiler_lib_search_path, $1)}" | $SED
-e 's! -L! !g' -e 's!^ !!'`
fi
_LT_TAGDECL([], [compiler_lib_search_dirs], [1],
   [The directories searched by this compiler when creating a shared library])
_LT_TAGDECL([], [predep_objects], [1],
   [Dependencies to place before and after the objects being linked to
   create a shared library])
_LT_TAGDECL([], [postdep_objects], [1])
_LT_TAGDECL([], [predeps], [1])
_LT_TAGDECL([], [postdeps], [1])
_LT_TAGDECL([], [compiler_lib_search_path], [1],
   [The library search path used internally by the compiler when linking
   a shared library])
])# _LT_SYS_HIDDEN_LIBDEPS


# _LT_LANG_F77_CONFIG([TAG])
# --------------------------
# Ensure that the configuration variables for a Fortran 77 compiler are
# suitably defined.  These variables are subsequently used by _LT_CONFIG
# to write the compiler configuration to 'libtool'.
m4_defun([_LT_LANG_F77_CONFIG],
[AC_LANG_PUSH(Fortran 77)
if test -z "$F77" || test no = "$F77"; then
```

```
 _lt_disable_F77=yes
fi

_LT_TAGVAR(archive_cmds_need_lc, $1)=no
_LT_TAGVAR(allow_undefined_flag, $1)=
_LT_TAGVAR(always_export_symbols, $1)=no
_LT_TAGVAR(archive_expsym_cmds, $1)=
_LT_TAGVAR(export_dynamic_flag_spec, $1)=
_LT_TAGVAR(hardcode_direct, $1)=no
_LT_TAGVAR(hardcode_direct_absolute, $1)=no
_LT_TAGVAR(hardcode_libdir_flag_spec, $1)=
_LT_TAGVAR(hardcode_libdir_separator, $1)=
_LT_TAGVAR(hardcode_minus_L, $1)=no
_LT_TAGVAR(hardcode_automatic, $1)=no
_LT_TAGVAR(inherit_rpath, $1)=no
_LT_TAGVAR(module_cmds, $1)=
_LT_TAGVAR(module_expsym_cmds, $1)=
_LT_TAGVAR(link_all_deplibs, $1)=unknown
_LT_TAGVAR(old_archive_cmds, $1)=$old_archive_cmds
_LT_TAGVAR(reload_flag, $1)=$reload_flag
_LT_TAGVAR(reload_cmds, $1)=$reload_cmds
_LT_TAGVAR(no_undefined_flag, $1)=
_LT_TAGVAR(whole_archive_flag_spec, $1)=
_LT_TAGVAR(enable_shared_with_static_runtimes, $1)=no

# Source file extension for f77 test sources.
ac_ext=f

# Object file extension for compiled f77 test sources.
objext=o
_LT_TAGVAR(objext, $1)=$objext

# No sense in running all these tests if we already determined that
# the F77 compiler isn't working.  Some variables (like enable_shared)
# are currently assumed to apply to all compilers on this platform,
# and will be corrupted by setting them based on a non-working compiler.
if test yes != "$_lt_disable_F77"; then
 # Code to be used in simple compile tests
 lt_simple_compile_test_code="\
     subroutine t
     return
     end
"

 # Code to be used in simple link tests
 lt_simple_link_test_code="\
     program t
     end
```

```
"

# ltmain only uses $CC for tagged configurations so make sure $CC is set.
_LT_TAG_COMPILER

# save warnings/boilerplate of simple test code
_LT_COMPILER_BOILERPLATE
_LT_LINKER_BOILERPLATE

# Allow CC to be a program name with arguments.
lt_save_CC=$CC
lt_save_GCC=$GCC
lt_save_CFLAGS=$CFLAGS
CC=${F77-"f77"}
CFLAGS=$FFLAGS
compiler=$CC
_LT_TAGVAR(compiler, $1)=$CC
_LT_CC_BASENAME([$compiler])
GCC=$G77
if test -n "$compiler"; then
  AC_MSG_CHECKING([if libtool supports shared libraries])
  AC_MSG_RESULT([$can_build_shared])

  AC_MSG_CHECKING([whether to build shared libraries])
  test no = "$can_build_shared" && enable_shared=no

  # On AIX, shared libraries and static libraries use the same namespace, and
  # are all built from PIC.
  case $host_os in
    aix3*)
      test yes = "$enable_shared" && enable_static=no
      if test -n "$RANLIB"; then
        archive_cmds="$archive_cmds~\$RANLIB \$lib"
        postinstall_cmds='$RANLIB $lib'
      fi
      ;;
    aix[[4-9]]*)
if test ia64 != "$host_cpu"; then
  case $enable_shared,$with_aix_soname,$aix_use_runtimelinking in
  yes,aix,yes) ;;  # shared object as lib.so file only
  yes,svr4,*) ;;  # shared object as lib.so archive member only
  yes,*) enable_static=no ;; # shared object in lib.a archive as well
  esac
fi
      ;;
  esac
  AC_MSG_RESULT([$enable_shared])
```

```
   AC_MSG_CHECKING([whether to build static libraries])
   # Make sure either enable_shared or enable_static is yes.
   test yes = "$enable_shared" || enable_static=yes
   AC_MSG_RESULT([$enable_static])

   _LT_TAGVAR(GCC, $1)=$G77
   _LT_TAGVAR(LD, $1)=$LD

   ## CAVEAT EMPTOR:
   ## There is no encapsulation within the following macros, do not change
   ## the running order or otherwise move them around unless you know exactly
   ## what you are doing...
   _LT_COMPILER_PIC($1)
   _LT_COMPILER_C_O($1)
   _LT_COMPILER_FILE_LOCKS($1)
   _LT_LINKER_SHLIBS($1)
   _LT_SYS_DYNAMIC_LINKER($1)
   _LT_LINKER_HARDCODE_LIBPATH($1)

   _LT_CONFIG($1)
 fi # test -n "$compiler"

 GCC=$lt_save_GCC
 CC=$lt_save_CC
 CFLAGS=$lt_save_CFLAGS
fi # test yes != "$_lt_disable_F77"

AC_LANG_POP
])# _LT_LANG_F77_CONFIG


# _LT_LANG_FC_CONFIG([TAG])
# -------------------------
# Ensure that the configuration variables for a Fortran compiler are
# suitably defined.  These variables are subsequently used by _LT_CONFIG
# to write the compiler configuration to 'libtool'.
m4_defun([_LT_LANG_FC_CONFIG],
[AC_LANG_PUSH(Fortran)

if test -z "$FC" || test no = "$FC"; then
 _lt_disable_FC=yes
fi

_LT_TAGVAR(archive_cmds_need_lc, $1)=no
_LT_TAGVAR(allow_undefined_flag, $1)=
_LT_TAGVAR(always_export_symbols, $1)=no
_LT_TAGVAR(archive_expsym_cmds, $1)=
_LT_TAGVAR(export_dynamic_flag_spec, $1)=
_LT_TAGVAR(hardcode_direct, $1)=no
_LT_TAGVAR(hardcode_direct_absolute, $1)=no
_LT_TAGVAR(hardcode_libdir_flag_spec, $1)=
_LT_TAGVAR(hardcode_libdir_separator, $1)=
_LT_TAGVAR(hardcode_minus_L, $1)=no
_LT_TAGVAR(hardcode_automatic, $1)=no
_LT_TAGVAR(inherit_rpath, $1)=no
_LT_TAGVAR(module_cmds, $1)=
_LT_TAGVAR(module_expsym_cmds, $1)=
_LT_TAGVAR(link_all_deplibs, $1)=unknown
_LT_TAGVAR(old_archive_cmds, $1)=$old_archive_cmds
_LT_TAGVAR(reload_flag, $1)=$reload_flag
_LT_TAGVAR(reload_cmds, $1)=$reload_cmds
_LT_TAGVAR(no_undefined_flag, $1)=
_LT_TAGVAR(whole_archive_flag_spec, $1)=
_LT_TAGVAR(enable_shared_with_static_runtimes, $1)=no

# Source file extension for fc test sources.
ac_ext=${ac_fc_srcext-f}

# Object file extension for compiled fc test sources.
objext=o
_LT_TAGVAR(objext, $1)=$objext

# No sense in running all these tests if we already determined that
# the FC compiler isn't working.  Some variables (like enable_shared)
# are currently assumed to apply to all compilers on this platform,
# and will be corrupted by setting them based on a non-working compiler.
if test yes != "$_lt_disable_FC"; then
  # Code to be used in simple compile tests
  lt_simple_compile_test_code="\
      subroutine t
      return
      end
"

  # Code to be used in simple link tests
  lt_simple_link_test_code="\
      program t
      end
"

  # ltmain only uses $CC for tagged configurations so make sure $CC is set.
  _LT_TAG_COMPILER

  # save warnings/boilerplate of simple test code
  _LT_COMPILER_BOILERPLATE
  _LT_LINKER_BOILERPLATE
```

```
# Allow CC to be a program name with arguments.
lt_save_CC=$CC
lt_save_GCC=$GCC
lt_save_CFLAGS=$CFLAGS
CC=${FC-"f95"}
CFLAGS=$FCFLAGS
compiler=$CC
GCC=$ac_cv_fc_compiler_gnu

_LT_TAGVAR(compiler, $1)=$CC
_LT_CC_BASENAME([$compiler])

if test -n "$compiler"; then
  AC_MSG_CHECKING([if libtool supports shared libraries])
  AC_MSG_RESULT([$can_build_shared])

  AC_MSG_CHECKING([whether to build shared libraries])
  test no = "$can_build_shared" && enable_shared=no

  # On AIX, shared libraries and static libraries use the same namespace, and
  # are all built from PIC.
  case $host_os in
    aix3*)
      test yes = "$enable_shared" && enable_static=no
      if test -n "$RANLIB"; then
        archive_cmds="$archive_cmds~\$RANLIB \$lib"
        postinstall_cmds='$RANLIB $lib'
      fi
      ;;
    aix[[4-9]]*)
if test ia64 != "$host_cpu"; then
  case $enable_shared,$with_aix_soname,$aix_use_runtimelinking in
  yes,aix,yes) ;;  # shared object as lib.so file only
  yes,svr4,*) ;;  # shared object as lib.so archive member only
  yes,*) enable_static=no ;; # shared object in lib.a archive as well
  esac
fi
      ;;
  esac
  AC_MSG_RESULT([$enable_shared])

  AC_MSG_CHECKING([whether to build static libraries])
  # Make sure either enable_shared or enable_static is yes.
  test yes = "$enable_shared" || enable_static=yes
  AC_MSG_RESULT([$enable_static])

  _LT_TAGVAR(GCC, $1)=$ac_cv_fc_compiler_gnu
```

```
    _LT_TAGVAR(LD, $1)=$LD

    ## CAVEAT EMPTOR:
    ## There is no encapsulation within the following macros, do not change
    ## the running order or otherwise move them around unless you know exactly
    ## what you are doing...
    _LT_SYS_HIDDEN_LIBDEPS($1)
    _LT_COMPILER_PIC($1)
    _LT_COMPILER_C_O($1)
    _LT_COMPILER_FILE_LOCKS($1)
    _LT_LINKER_SHLIBS($1)
    _LT_SYS_DYNAMIC_LINKER($1)
    _LT_LINKER_HARDCODE_LIBPATH($1)

    _LT_CONFIG($1)
  fi # test -n "$compiler"

  GCC=$lt_save_GCC
  CC=$lt_save_CC
  CFLAGS=$lt_save_CFLAGS
fi # test yes != "$_lt_disable_FC"

AC_LANG_POP
])# _LT_LANG_FC_CONFIG


# _LT_LANG_GCJ_CONFIG([TAG])
# --------------------------
# Ensure that the configuration variables for the GNU Java Compiler compiler
# are suitably defined.  These variables are subsequently used by _LT_CONFIG
# to write the compiler configuration to 'libtool'.
m4_defun([_LT_LANG_GCJ_CONFIG],
[AC_REQUIRE([LT_PROG_GCJ])dnl
AC_LANG_SAVE

# Source file extension for Java test sources.
ac_ext=java

# Object file extension for compiled Java test sources.
objext=o
_LT_TAGVAR(objext, $1)=$objext

# Code to be used in simple compile tests
lt_simple_compile_test_code="class foo {}"

# Code to be used in simple link tests
lt_simple_link_test_code='public class conftest { public static void main(String[[]] argv) {}; }'
```

```
# ltmain only uses $CC for tagged configurations so make sure $CC is set.
_LT_TAG_COMPILER

# save warnings/boilerplate of simple test code
_LT_COMPILER_BOILERPLATE
_LT_LINKER_BOILERPLATE

# Allow CC to be a program name with arguments.
lt_save_CC=$CC
lt_save_CFLAGS=$CFLAGS
lt_save_GCC=$GCC
GCC=yes
CC=${GCJ-"gcj"}
CFLAGS=$GCJFLAGS
compiler=$CC
_LT_TAGVAR(compiler, $1)=$CC
_LT_TAGVAR(LD, $1)=$LD
_LT_CC_BASENAME([$compiler])

# GCJ did not exist at the time GCC didn't implicitly link libc in.
_LT_TAGVAR(archive_cmds_need_lc, $1)=no

_LT_TAGVAR(old_archive_cmds, $1)=$old_archive_cmds
_LT_TAGVAR(reload_flag, $1)=$reload_flag
_LT_TAGVAR(reload_cmds, $1)=$reload_cmds

## CAVEAT EMPTOR:
## There is no encapsulation within the following macros, do not change
## the running order or otherwise move them around unless you know exactly
## what you are doing...
if test -n "$compiler"; then
 _LT_COMPILER_NO_RTTI($1)
 _LT_COMPILER_PIC($1)
 _LT_COMPILER_C_O($1)
 _LT_COMPILER_FILE_LOCKS($1)
 _LT_LINKER_SHLIBS($1)
 _LT_LINKER_HARDCODE_LIBPATH($1)

 _LT_CONFIG($1)
fi

AC_LANG_RESTORE

GCC=$lt_save_GCC
CC=$lt_save_CC
CFLAGS=$lt_save_CFLAGS
])# _LT_LANG_GCJ_CONFIG
```

```
# _LT_LANG_GO_CONFIG([TAG])
# --------------------------
# Ensure that the configuration variables for the GNU Go compiler
# are suitably defined.  These variables are subsequently used by _LT_CONFIG
# to write the compiler configuration to 'libtool'.
m4_defun([_LT_LANG_GO_CONFIG],
[AC_REQUIRE([LT_PROG_GO])dnl
AC_LANG_SAVE

# Source file extension for Go test sources.
ac_ext=go

# Object file extension for compiled Go test sources.
objext=o
_LT_TAGVAR(objext, $1)=$objext

# Code to be used in simple compile tests
lt_simple_compile_test_code="package main; func main() { }"

# Code to be used in simple link tests
lt_simple_link_test_code='package main; func main() { }'

# ltmain only uses $CC for tagged configurations so make sure $CC is set.
_LT_TAG_COMPILER

# save warnings/boilerplate of simple test code
_LT_COMPILER_BOILERPLATE
_LT_LINKER_BOILERPLATE

# Allow CC to be a program name with arguments.
lt_save_CC=$CC
lt_save_CFLAGS=$CFLAGS
lt_save_GCC=$GCC
GCC=yes
CC=${GOC-"gccgo"}
CFLAGS=$GOFLAGS
compiler=$CC
_LT_TAGVAR(compiler, $1)=$CC
_LT_TAGVAR(LD, $1)=$LD
_LT_CC_BASENAME([$compiler])

# Go did not exist at the time GCC didn't implicitly link libc in.
_LT_TAGVAR(archive_cmds_need_lc, $1)=no

_LT_TAGVAR(old_archive_cmds, $1)=$old_archive_cmds
_LT_TAGVAR(reload_flag, $1)=$reload_flag
_LT_TAGVAR(reload_cmds, $1)=$reload_cmds
```

```
## CAVEAT EMPTOR:
## There is no encapsulation within the following macros, do not change
## the running order or otherwise move them around unless you know exactly
## what you are doing...
if test -n "$compiler"; then
 _LT_COMPILER_NO_RTTI($1)
 _LT_COMPILER_PIC($1)
 _LT_COMPILER_C_O($1)
 _LT_COMPILER_FILE_LOCKS($1)
 _LT_LINKER_SHLIBS($1)
 _LT_LINKER_HARDCODE_LIBPATH($1)

 _LT_CONFIG($1)
fi

AC_LANG_RESTORE

GCC=$lt_save_GCC
CC=$lt_save_CC
CFLAGS=$lt_save_CFLAGS
])# _LT_LANG_GO_CONFIG


# _LT_LANG_RC_CONFIG([TAG])
# -------------------------
# Ensure that the configuration variables for the Windows resource compiler
# are suitably defined.  These variables are subsequently used by _LT_CONFIG
# to write the compiler configuration to 'libtool'.
m4_defun([_LT_LANG_RC_CONFIG],
[AC_REQUIRE([LT_PROG_RC])dnl
AC_LANG_SAVE

# Source file extension for RC test sources.
ac_ext=rc

# Object file extension for compiled RC test sources.
objext=o
_LT_TAGVAR(objext, $1)=$objext

# Code to be used in simple compile tests
lt_simple_compile_test_code='sample MENU { MENUITEM "&Soup", 100, CHECKED }'

# Code to be used in simple link tests
lt_simple_link_test_code=$lt_simple_compile_test_code

# ltmain only uses $CC for tagged configurations so make sure $CC is set.
_LT_TAG_COMPILER
```

```
# save warnings/boilerplate of simple test code
_LT_COMPILER_BOILERPLATE
_LT_LINKER_BOILERPLATE

# Allow CC to be a program name with arguments.
lt_save_CC=$CC
lt_save_CFLAGS=$CFLAGS
lt_save_GCC=$GCC
GCC=
CC=${RC-"windres"}
CFLAGS=
compiler=$CC
_LT_TAGVAR(compiler, $1)=$CC
_LT_CC_BASENAME([$compiler])
_LT_TAGVAR(lt_cv_prog_compiler_c_o, $1)=yes

if test -n "$compiler"; then
 :
 _LT_CONFIG($1)
fi

GCC=$lt_save_GCC
AC_LANG_RESTORE
CC=$lt_save_CC
CFLAGS=$lt_save_CFLAGS
])# _LT_LANG_RC_CONFIG


# LT_PROG_GCJ
# -----------
AC_DEFUN([LT_PROG_GCJ],
[m4_ifdef([AC_PROG_GCJ], [AC_PROG_GCJ],
 [m4_ifdef([A][M_PROG_GCJ], [A][M_PROG_GCJ],
   [AC_CHECK_TOOL(GCJ, gcj,)
     test set = "${GCJFLAGS+set}" || GCJFLAGS="-g -O2"
     AC_SUBST(GCJFLAGS)])])[]dnl
])

# Old name:
AU_ALIAS([LT_AC_PROG_GCJ], [LT_PROG_GCJ])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([LT_AC_PROG_GCJ], [])


# LT_PROG_GO
# ----------
AC_DEFUN([LT_PROG_GO],
[AC_CHECK_TOOL(GOC, gccgo,)
])


# LT_PROG_RC
# ----------
AC_DEFUN([LT_PROG_RC],
[AC_CHECK_TOOL(RC, windres,)
])

# Old name:
AU_ALIAS([LT_AC_PROG_RC], [LT_PROG_RC])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([LT_AC_PROG_RC], [])


# _LT_DECL_EGREP
# --------------
# If we don't have a new enough Autoconf to choose the best grep
# available, choose the one first in the user's PATH.
m4_defun([_LT_DECL_EGREP],
[AC_REQUIRE([AC_PROG_EGREP])dnl
AC_REQUIRE([AC_PROG_FGREP])dnl
test -z "$GREP" && GREP=grep
_LT_DECL([], [GREP], [1], [A grep program that handles long lines])
_LT_DECL([], [EGREP], [1], [An ERE matcher])
_LT_DECL([], [FGREP], [1], [A literal string matcher])
dnl Non-bleeding-edge autoconf doesn't subst GREP, so do it here too
AC_SUBST([GREP])
])


# _LT_DECL_OBJDUMP
# --------------
# If we don't have a new enough Autoconf to choose the best objdump
# available, choose the one first in the user's PATH.
m4_defun([_LT_DECL_OBJDUMP],
[AC_CHECK_TOOL(OBJDUMP, objdump, false)
test -z "$OBJDUMP" && OBJDUMP=objdump
_LT_DECL([], [OBJDUMP], [1], [An object symbol dumper])
AC_SUBST([OBJDUMP])
])

# _LT_DECL_DLLTOOL
# ----------------
# Ensure DLLTOOL variable is set.
m4_defun([_LT_DECL_DLLTOOL],
[AC_CHECK_TOOL(DLLTOOL, dlltool, false)
```

```
test -z "$DLLTOOL" && DLLTOOL=dlltool
_LT_DECL([], [DLLTOOL], [1], [DLL creation program])
AC_SUBST([DLLTOOL])
])

# _LT_DECL_SED
# ------------
# Check for a fully-functional sed program, that truncates
# as few characters as possible.  Prefer GNU sed if found.
m4_defun([_LT_DECL_SED],
[AC_PROG_SED
test -z "$SED" && SED=sed
Xsed="$SED -e 1s/^X//"
_LT_DECL([], [SED], [1], [A sed program that does not truncate output])
_LT_DECL([], [Xsed], ["\$SED -e 1s/^X//"],
   [Sed that helps us avoid accidentally triggering echo(1) options like -n])
])# _LT_DECL_SED

m4_ifndef([AC_PROG_SED], [
############################################################
# NOTE: This macro has been submitted for inclusion into   #
#  GNU Autoconf as AC_PROG_SED.  When it is available in   #
#  a released version of Autoconf we should remove this    #
#  macro and use it instead.                               #
############################################################

m4_defun([AC_PROG_SED],
[AC_MSG_CHECKING([for a sed that does not truncate output])
AC_CACHE_VAL(lt_cv_path_SED,
[# Loop through the user's path and test for sed and gsed.
# Then use that list of sed's as ones to test for truncation.
as_save_IFS=$IFS; IFS=$PATH_SEPARATOR
for as_dir in $PATH
do
 IFS=$as_save_IFS
 test -z "$as_dir" && as_dir=.
 for lt_ac_prog in sed gsed; do
   for ac_exec_ext in '' $ac_executable_extensions; do
     if $as_executable_p "$as_dir/$lt_ac_prog$ac_exec_ext"; then
       lt_ac_sed_list="$lt_ac_sed_list $as_dir/$lt_ac_prog$ac_exec_ext"
     fi
   done
 done
done
IFS=$as_save_IFS
lt_ac_max=0
lt_ac_count=0
# Add /usr/xpg4/bin/sed as it is typically found on Solaris
# along with /bin/sed that truncates output.
for lt_ac_sed in $lt_ac_sed_list /usr/xpg4/bin/sed; do
 test ! -f "$lt_ac_sed" && continue
 cat /dev/null > conftest.in
 lt_ac_count=0
 echo $ECHO_N "0123456789$ECHO_C" >conftest.in
 # Check for GNU sed and select it if it is found.
 if "$lt_ac_sed" --version 2>&1 < /dev/null | grep 'GNU' > /dev/null; then
   lt_cv_path_SED=$lt_ac_sed
   break
 fi
 while true; do
   cat conftest.in conftest.in >conftest.tmp
   mv -f conftest.tmp conftest.in
   cp conftest.in conftest.nl
   echo >>conftest.nl
   $lt_ac_sed -e 's/a$//' < conftest.nl >conftest.out || break
   cmp -s conftest.out conftest.nl || break
   # 10000 chars as input seems more than enough
   test 10 -lt "$lt_ac_count" && break
   lt_ac_count=`expr $lt_ac_count + 1`
   if test "$lt_ac_count" -gt "$lt_ac_max"; then
     lt_ac_max=$lt_ac_count
     lt_cv_path_SED=$lt_ac_sed
   fi
 done
done
])
SED=$lt_cv_path_SED
AC_SUBST([SED])
AC_MSG_RESULT([$SED])
])#AC_PROG_SED
])#m4_ifndef

# Old name:
AU_ALIAS([LT_AC_PROG_SED], [AC_PROG_SED])
dnl aclocal-1.4 backwards compatibility:
dnl AC_DEFUN([LT_AC_PROG_SED], [])


# _LT_CHECK_SHELL_FEATURES
# ------------------------
# Find out whether the shell is Bourne or XSI compatible,
# or has some other useful features.
m4_defun([_LT_CHECK_SHELL_FEATURES],
[if ( (MAIL=60; unset MAIL) || exit) >/dev/null 2>&1; then
 lt_unset=unset
else
```

```
 lt_unset=false
fi
_LT_DECL([], [lt_unset], [0], [whether the shell understands "unset"])dnl

# test EBCDIC or ASCII
case `echo X|tr X '\101'` in
A) # ASCII based system
   # \n is not interpreted correctly by Solaris 8 /usr/ucb/tr
 lt_SP2NL='tr \040 \012'
 lt_NL2SP='tr \015\012 \040\040'
 ;;
*) # EBCDIC based system
 lt_SP2NL='tr \100 \n'
 lt_NL2SP='tr \r\n \100\100'
 ;;
esac
_LT_DECL([SP2NL], [lt_SP2NL], [1], [turn spaces into newlines])dnl
_LT_DECL([NL2SP], [lt_NL2SP], [1], [turn newlines into spaces])dnl
])# _LT_CHECK_SHELL_FEATURES


# _LT_PATH_CONVERSION_FUNCTIONS
# -----------------------------
# Determine what file name conversion functions should be used by
# func_to_host_file (and, implicitly, by func_to_host_path).  These are needed
# for certain cross-compile configurations and native mingw.
m4_defun([_LT_PATH_CONVERSION_FUNCTIONS],
[AC_REQUIRE([AC_CANONICAL_HOST])dnl
AC_REQUIRE([AC_CANONICAL_BUILD])dnl
AC_MSG_CHECKING([how to convert $build file names to $host format])
AC_CACHE_VAL(lt_cv_to_host_file_cmd,
[case $host in
 *-*-mingw* )
   case $build in
     *-*-mingw* ) # actually msys
       lt_cv_to_host_file_cmd=func_convert_file_msys_to_w32
       ;;
     *-*-cygwin* )
       lt_cv_to_host_file_cmd=func_convert_file_cygwin_to_w32
       ;;
     * ) # otherwise, assume *nix
       lt_cv_to_host_file_cmd=func_convert_file_nix_to_w32
       ;;
   esac
   ;;
 *-*-cygwin* )
   case $build in
     *-*-mingw* ) # actually msys
```

```
      lt_cv_to_host_file_cmd=func_convert_file_msys_to_cygwin
      ;;
    *-*-cygwin* )
      lt_cv_to_host_file_cmd=func_convert_file_noop
      ;;
    * ) # otherwise, assume *nix
      lt_cv_to_host_file_cmd=func_convert_file_nix_to_cygwin
      ;;
  esac
  ;;
 * ) # unhandled hosts (and "normal" native builds)
  lt_cv_to_host_file_cmd=func_convert_file_noop
  ;;
esac
])
to_host_file_cmd=$lt_cv_to_host_file_cmd
AC_MSG_RESULT([$lt_cv_to_host_file_cmd])
_LT_DECL([to_host_file_cmd], [lt_cv_to_host_file_cmd],
        [0], [convert $build file names to $host format])dnl

AC_MSG_CHECKING([how to convert $build file names to toolchain format])
AC_CACHE_VAL(lt_cv_to_tool_file_cmd,
[#assume ordinary cross tools, or native build.
lt_cv_to_tool_file_cmd=func_convert_file_noop
case $host in
 *-*-mingw* )
  case $build in
    *-*-mingw* ) # actually msys
      lt_cv_to_tool_file_cmd=func_convert_file_msys_to_w32
      ;;
  esac
  ;;
esac
])
to_tool_file_cmd=$lt_cv_to_tool_file_cmd
AC_MSG_RESULT([$lt_cv_to_tool_file_cmd])
_LT_DECL([to_tool_file_cmd], [lt_cv_to_tool_file_cmd],
        [0], [convert $build files to toolchain format])dnl
])# _LT_PATH_CONVERSION_FUNCTIONS
```

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the

manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

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 that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version"

applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.

1.56 librdkafka 0.11.6

1.56.1 Available under license :

From https://github.com/tinycthread/tinycthread/README.txt c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
2013-2014 Evan Nemerson

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.

The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.
==

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

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.
This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram https://github.com/codahale/hdrhistogram
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License (MIT)

Copyright (c) 2014 Coda Hale

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
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: librdkafka
Source: https://github.com/edenhill/librdkafka

License: BSD-2-clause
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 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.

Files: *
Copyright: 2012-2015, Magnus Edenhill
License: BSD-2-clause

Files: src/rdcrc32.c src/rdcrc32.h
Copyright: 2006-2012, Thomas Pircher <tehpeh@gmx.net>
License: MIT
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.

Files: src/snappy.c src/snappy.h src/snappy_compat.h
Copyright: 2005, Google Inc.
2011, Intel Corporation
License: BSD-3-clause
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.

Files: debian/*
Copyright: 2013 Faidon Liambotis <paravoid@debian.org>
License: BSD-2-clause
regexp.c and regexp.h from https://github.com/ccxvii/minilibs sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.
"
librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2018, Magnus Edenhill
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.

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.
LICENSE
----
librdkafka - Apache Kafka C driver library

Copyright (c) 2012-2018, Magnus Edenhill
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.

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.

LICENSE.crc32c

# For src/crc32c.c copied (with modifications) from

```
# http://stackoverflow.com/a/17646775/1821055

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1  1 Aug 2013  Mark Adler
*/

/*
 This software 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 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.

 Mark Adler
 madler@alumni.caltech.edu
*/
```

LICENSE.hdrhistogram

This license covers src/rdhdrhistogram.c which is a C port of
Coda Hale's Golang HdrHistogram https://github.com/codahale/hdrhistogram
at revision 3a0bb77429bd3a61596f5e8a3172445844342120

The MIT License (MIT)

Copyright (c) 2014 Coda Hale

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

LICENSE.lz4

----

src/xxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git e2827775ee80d2ef985858727575df31fc60f1f3

LZ4 Library
Copyright (c) 2011-2016, Yann Collet
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 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.

LICENSE.murmur2

----

parts of src/rdmurmur2.c: git@github.com:abrandoned/murmur2.git

MurMurHash2 Library

//---
// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

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.

LICENSE.pycrc

The following license applies to the files rdcrc32.c and rdcrc32.h which
have been generated by the pycrc tool.
==

Copyright (c) 2006-2012, Thomas Pircher <tehpeh@gmx.net>

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.

LICENSE.queue

For sys/queue.h:

* Copyright (c) 1991, 1993
* The Regents of the University of California. 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.
* 4. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* $FreeBSD$

LICENSE.regexp

----

regexp.c and regexp.h from https://github.com/ccxvii/minilibs sha 875c33568b5a4aa4fb3dd0c52ea98f7f0e5ca684

"

These libraries are in the public domain (or the equivalent where that is not possible). You can do anything you want with them. You have no legal obligation to do anything else, although I appreciate attribution.

"

LICENSE.snappy

```
########################################################################
# LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h   #
# originally retrieved from http://github.com/andikleen/snappy-c     #
# git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219              #
########################################################################
```

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation 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 Intel Corporation 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.

LICENSE.tinycthread

From https://github.com/tinycthread/tinycthread/README.txt c57166cd510ffb5022dd5f127489b131b61441b9

License

Copyright (c) 2012 Marcus Geelnard
           2013-2014 Evan Nemerson

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.

LICENSE.wingetopt

For the files wingetopt.c wingetopt.h downloaded from https://github.com/alex85k/wingetopt

```
/*
* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
*
* 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" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, 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.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*-
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* 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 NETBSD FOUNDATION, INC. 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 FOUNDATION 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.
*/
```

For the files wingetopt.c wingetopt.h downloaded from https://github.com/alex85k/wingetopt

```
/*
* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>
*
* 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" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, 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.
*
* Sponsored in part by the Defense Advanced Research Projects
* Agency (DARPA) and Air Force Research Laboratory, Air Force
* Materiel Command, USAF, under agreement number F39502-99-1-0512.
*/
/*-
* Copyright (c) 2000 The NetBSD Foundation, Inc.
* All rights reserved.
*
```

```
* This code is derived from software contributed to The NetBSD Foundation
* by Dieter Baron and Thomas Klausner.
*
* 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 NETBSD FOUNDATION, INC. 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 FOUNDATION 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.
*/
# For src/crc32c.c copied (with modifications) from
# http://stackoverflow.com/a/17646775/1821055

/* crc32c.c -- compute CRC-32C using the Intel crc32 instruction
* Copyright (C) 2013 Mark Adler
* Version 1.1  1 Aug 2013  Mark Adler
*/

/*
 This software 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 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.
```

 Mark Adler
 madler@alumni.caltech.edu
*/
For sys/queue.h:

* Copyright (c) 1991, 1993
* The Regents of the University of California. 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.
* 4. Neither the name of the University nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)queue.h 8.5 (Berkeley) 8/20/94
* $FreeBSD$
##
# LICENSE.snappy covers files: snappy.c, snappy.h, snappy_compat.h   #
# originally retrieved from http://github.com/andikleen/snappy-c   #
# git revision 8015f2d28739b9a6076ebaa6c53fe27bc238d219   #
##

The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation 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 Intel Corporation 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.
src/xxhash.[ch] src/lz4*.[ch]: git@github.com:lz4/lz4.git e2827775ee80d2ef985858727575df31fc60f1f3

LZ4 Library

Copyright (c) 2011-2016, Yann Collet
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 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.

parts of src/rdmurmur2.c: git@github.com:abrandoned/murmur2.git

MurMurHash2 Library
//---
// MurmurHash2 was written by Austin Appleby, and is placed in the public
// domain. The author hereby disclaims copyright to this source code.

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.

1.57 libseccomp 2.3.1 :3.el7

1.57.1 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.

1.58 libseccomp 2.3.1 :3.el7

1.58.1 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.

1.59 libsodium 1.0.16 :1.0

1.59.1 Available under license :

/*
* ISC License
*
* Copyright (c) 2013-2017
* Frank Denis <j at pureftpd dot org>
*
* Permission to use, copy, modify, and/or 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" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
* ANY SPECIAL, DIRECT, 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.
*/

1.60 libzmq 4.3.2

1.60.1 Available under license :

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Piotr Trojanek that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "ptroja", with commit author "Piotr Trojanek <piotr.trojanek@gmail.com>", are copyright of Piotr Trojanek. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Piotr Trojanek
2017/03/19
# Permission to Relicense under MPLv2 or any other OSI approved license chosen
# by the current ZeroMQ BDFL

This is a statement by Nikolay Amiantov that grants permission to relicense his copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "abbradar", with commit author "Nikolay Amiantov <ab@fmap.me>", are copyright of Nikolay Amiantov. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Nikolay Amiantov
2017/03/24
Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by "Michal Vyskocil" that grants permission to relicense its copyrights in the libzmq C++ library

(ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "vyskocilm", with commit author "Michal Vyskocil michal.vyskocil@gmail.com", are copyright of "Michal Vyskocil". This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Michal Vyskocil 2017/03/24

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Matthias Kluwe
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "mkluwe", with
commit author "Matthias Kluwe <mkluwe@gmail.com>", are copyright of Matthias Kluwe.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Matthias Kluwe
2017/03/24

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Fabrice Fontaine
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "ffontaine", with
commit author "Fabrice Fontaine <fontaine.fabrice@gmail.com>", are copyright of Fabrice Fontaine.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Fabrice Fontaine
2018-09-13

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Leonardo J. Consoni
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "Bitiquinho", with
commit author "Leonardo Jos Consoni <consoni_2519@hotmail.com>", are copyright of Leonardo J. Consoni.

This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Leonardo Jos Consoni
2017/03/19
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Daniel Shih
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "danielhtshih", with
commit author "Daniel Shih <hotingwow@gmail.com>", are copyright of Daniel Shih.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Daniel Shih
2017/03/20
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Ilya Kulakov
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "kentzo", with
commit author "Ilya Kulakov <kulakov.ilya@gmail.com>", are copyright of Ilya Kulakov.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Ilya Kulakov
2017/05/23
# Permission to Relicense under MPLv2 or any other OSI approved license chosen
# by the current ZeroMQ BDFL

This is a statement by Wouter Overmeire that grants permission to relicense its
copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License
v2 (MPLv2) or any other Open Source Initiative approved license chosen by the
current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "lodagro", with commit
author "lodagro <lodagro@gmail.com>", are copyright of Wouter Overmeire. This
document hereby grants the libzmq project team to relicense libzmq, including
all past, present and future contributions of the author listed above.

Wouter Overmeire
2017/04/01

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Dan Riegsecker that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "danriegsecker", with commit author "Dan Riegsecker 1baldgeek@gmail.com", are copyright of Dan Riegsecker. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Dan Riegsecker
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Roy Lenferink that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "rlenferink", with commit author "Roy Lenferink <lenferinkroy@gmail.com>" or "Roy Lenferink <rlenferink@apache.org>", are copyright of Roy Lenferink.
This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Roy Lenferink
2017/03/17

# Permission to Relicense under MPLv2

This is a statement by Kapp Arnaud that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "xaqq", with commit author "Kapp Arnaud", are copyright of Kapp Arnaud.
This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Kapp Arnaud
2017/03/30

# Permission to Relicense under MPLv2

This is a statement by IVU Traffic Technologies AG
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "jruffin", with
commit author "Julien Ruffin <jur@ivu.de>", are copyright of IVU Traffic Technologies AG.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Henrik Benner
Head of Department Ticketing Development, IVU Traffic Technologies AG
2017/05/10

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Giuseppe Corbelli
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "cowo78" with
commit author "Giuseppe Corbelli <cowo78@gmail.com>", are copyright of Giuseppe Corbelli.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Giuseppe Corbelli
2017/03/21

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Eric Voskuil
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "evoskuil", with
commit author "Eric Voskuil <eric@voskuil.org>", are copyright of Eric Voskuil.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Eric Voskuil
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Florian Ebeling that grants permission to
relicense its copyrights in the libzmq C++ library (ZeroMQ) under the

Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "febeling", with commit author "Florian Ebeling <florian.ebeling@gmail.com>", are copyright of Florian Ebeling. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Florian Ebeling
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Constantin Rack that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "c-rack", with commit author "Constantin Rack <constantin.rack@gmail.com>", are copyright of Constantin Rack. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Constantin Rack
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Robert Gagnon that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "rgagnon24", with commit author "Robert Gagnon rgagnon24+zmq@gmail.com", are copyright of Robert Gagnon. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Robert Gagnon
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Min Ragan-Kelley that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the GitHub handle "minrk", with
commit author "Min RK benjaminrk@gmail.com", are copyright of Min Ragan-Kelley.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Min Ragan-Kelley
2017/07/17
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Thomas Rodgers
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "rodgert", with
commit author "Thomas Rodgers <rodgert@twrodgers.com>", are copyright
of Thomas Rodgers.

This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Thomas W Rodgers
2017/04/08
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Matthew Arsenault
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "arsenm", with
commit author "Matt Arsenault <arsenm2@gmail.com>", are copyright of Matthew Arsenault.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Matthew Arsenault
2017/10/12
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Reza Ebrahimi
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "reza-ebrahimi", with

commit author "Reza Ebrahimi <reza.ebrahimi.dev@gmail.com>", are copyright of Reza Ebrahimi.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Reza Ebrahimi
2017/03/22
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Egomotion Limited
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "robertcastle", with
commit author "Robert Castle <robert@egomotion.co.uk>", are copyright of Egomotion Limited.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Robert Castle,
Director, Egomotion Limited
2017/03/21
# Permission to Relicense under MPLv2

This is a statement by Fedor Sheremetyev
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "sheremetyev", with
commit author "Fedor Sheremetyev <sheremetyev@gmail.com>", are copyright of Fedor Sheremetyev.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Fedor Sheremetyev
2017/03/20
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Jim Hague that grants permission to relicense
its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla
Public License v2 (MPLv2) or any other Open Source Initiative approved
license chosen by the current ZeroMQ BDFL (Benevolent Dictator for
Life).

A portion of the commits made by the Github handle "banburybill", with
commit author "Jim Hague <jim.hague@acm.org>", are copyright of Jim
Hague. This document hereby grants the libzmq project team to
relicense libzmq, including all past, present and future contributions
of the author listed above.

Jim Hague
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Justin.Hung
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "loachfish", with
commit author "Justin.Hung fronront@qq.com", are copyright of Justin.Hung .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Justin.Hung
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Lionel Orry
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "chickenkiler", with
commit author "Lionel Orry <lionel.orry@gmail.com>", are copyright of Lionel Orry.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Lionel Orry
2017/03/23

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Gavin McNiff
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "gavinmcniff", with
commit author "Gavin McNiff <gavin@mcniff.ie>", are copyright of Gavin McNiff .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Gavin McNiff
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Sylvain Corlay
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "SylvainCorlay", with
commit author "Sylvain Corlay sylvain.corlay@gmail.com", are copyright of Sylvain Corlay .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Sylvain Corlay
2017/03/20

# Permission to Relicense under MPLv2

This is a statement by Maurizio Melato
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "mauri-melato", with
commit author "Maurizio Melato", are copyright of Maurizio Melato.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Maurizio Melato
2017-04-04

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Adrian Muraru
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "rlenferink", with
commit author "Adrian Muraru <adi.muraru@gmail.com>", are copyright of Adrian Muraru.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Adrian Muraru
2017/03/20

# Permission to Relicense under MPLv2

This is a statement by Hendrik Beskow
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "Asmod4n", with
commit author "Asmod4n Asmod4n@users.noreply.github.com", are copyright of Hendrik Beskow.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Hendrik Beskow
2017/04/21

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Nathan Toone that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "toonetown", with commit author "Nathan Toone <nathan@toonetown.com>", are copyright of Nathan Toone. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Nathan Toone
2018/11/05

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by "Osiris Pedroso" that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "opedroso", with
commit author "Osiris Pedroso <opedroso@gmail.com>", are copyright of "Osiris Pedroso".
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Osiris Pedroso
2017/03/21

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Diego Fons that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "dfons", with

commit author "Diego Fons <diego.fons@gmail.com>" or "Diego Fons <diego.fons@intraway.com>", are copyright of Diego Fons.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Diego Fons

2017/03/20
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Maarten Ditzel
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "mditzel", with
commit author "Maarten Ditzel <maarten.ditzel@gmail.com>", are copyright of Maarten Ditzel.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Maarten Ditzel
2017/03/24
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Ian Barber
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commit made by the Github handle "ianbarber", with
commit author "Ian Barber <ianbarber@gmail.com>", are copyright of Ian Barber.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Ian Barber
2017/03/27
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Christian Kamm
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "ckamm", with
commit author "Christian Kamm kamm@incasoftware.de", are copyright of Christian Kamm.

This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Christian Kamm
2017/03/20

# Permission to Relicense under MPLv2

This is a statement by Brian Knox
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "taotetek", with
commit author "Brian Knox", are copyright of Brian Knox.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Brian Knox
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Brian Buchanan
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "bwbuchanan", with
commit author "Brian Buchanan <bwb@holo.org>" or "Brian Buchanan <brian@tiogalake.com>", are copyright of
Brian Buchanan.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Brian Buchanan
2017/03/23

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Bjrn Tpel that grants permission to relicense
its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla
Public License v2 (MPLv2) or any other Open Source Initiative approved
license chosen by the current ZeroMQ BDFL (Benevolent Dictator for
Life).

A portion of the commits made by the Github handle "bjoto", with
commit author "Bjorn Topel <bjorn.topel@gmail.com>", are copyright of
Bjrn Tpel. This document hereby grants the libzmq project team to
relicense libzmq, including all past, present and future contributions
of the author listed above.

Bjrn Tpel
2017/03/26

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Takeshi Abe
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "tabe", with
commit author "Takeshi Abe <tabe@fixedpoint.jp>", are copyright of Takeshi Abe.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Takeshi Abe
2017/03/22

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Andrew Thompson
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "thompsa", with
commit author "Andew Thompson <andy@fud.org.nz>", are copyright of Andrew Thompson.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Andrew Thompson
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Eric Camachat
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "camachat", with
commit author "Eric Camachat <eric@camachat.org>", are copyright of Eric Camachat.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Eric L. Camachat
March 20, 2017

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Elliot Saba
that grants permission to relicense his copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "staticfloat", with
commit author "Elliot Saba <staticfloat@gmail.com>", are copyright of Elliot Saba.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Elliot Saba
2017/03/19

# Permission to Relicense under MPLv2

This is a statement by Jake Cobb
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "jakecobb", with
commit author "Jake Cobb <jake.cobb@gmail.com>", are copyright of Jake Cobb.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Jake Cobb
2017/04/04

# Permission to Relicense under MPLv2

This is a statement by Floris Bruynooghe
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "flub", with
commit author "Floris Bruynooghe <flub@devork.be>", are copyright of
Floris Bruynooghe .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Floris Bruynooghe
2017/05/03

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Anton Dimitrov that grants permission to
relicense its copyrights in the libzmq C++ library (ZeroMQ) under the
Mozilla Public License v2 (MPLv2) or any other Open Source Initiative
approved license chosen by the current ZeroMQ BDFL (Benevolent

Dictator for Life).

A portion of the commits made by the Github handle "antonrd", with
commit author "Anton Dimitrov <dimitrov.anton@gmail.com>", are
copyright of Anton Dimitrov. This document hereby grants the libzmq
project team to relicense libzmq, including all past, present and
future contributions of the author listed above.

Anton Dimitrov

2018/10/01

# Permission to Relicense under MPLv2

This is a statement by Bruno Bodin
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "bbdb68", with
commit author "Bruno Bodin <brunobodin@gmail.com>", are copyright of Bruno Bodin .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Bruno Bodin
2017/04/04

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Gonzalo Diethelm
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "gonzus", with
commit author "Gonzalo Diethelm <gonzalo.diethelm@diethelm.org>" or "Gonzalo Diethelm
<gdiethelm@dcv.cl>", are copyright of Gonzalo Diethelm .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Gonzalo Diethelm
2017/03/20

# Permission to Relicense under MPLv2 or any other share-alike OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by {{ name of company / name of individual }}
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
share-alike Open Source Initiative approved license chosen by the current
ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "{{github username}}", with
commit author "{{github commit author}}", are copyright of {{ name }} .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

{{ Full Name }}
{{ creation date of document (format: yyyy/mm/dd) }}
# Permission to Relicense under MPLv2

This is a statement by {{ name of company / name of individual }}
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "{{github username}}", with
commit author "{{github commit author}}", are copyright of {{ name }} .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

{{ Full Name }}
{{ creation date of document (format: yyyy/mm/dd) }}
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by {{ name of company / name of individual }}
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "{{github username}}", with
commit author "{{github commit author}}", are copyright of {{ name }} .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

{{ Full Name }}
{{ creation date of document (format: yyyy/mm/dd) }}
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Anthony Scemama
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "scemama", with
commit author "Anthony Scemama <scemama@irsamc.ups-tlse.fr>", are copyright of Anthony Scemama.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Anthony Scemama
2017/03/21
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by William Swanson
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "swansontec", with
commit author " William Swanson <swansontec@gmail.com>", are copyright of William Swanson.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

William Swanson
2017/04/10
# Permission to Relicense under MPLv2

This is a statement by Google, Inc.
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "tkoeppe", with
commit author "Thomas Kppe", are copyright of Google, Inc.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Thomas Kppe
2017/09/13
# Permission to Relicense under MPLv2

This is a statement by Bernd Prager
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "bprager", with
commit author "Bernd Prager <bernd@prager.ws>", are copyright of Bernd Prager.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Bernd Prager
2017/03/22
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Harald Achitz
that grants permission to relicense its copyrights in the libzmq C++

library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "a4z", with
commit author "Harald Achitz <harald.achitz@gmail.com>" or
"Harald Achitz <harald.achitz@tritech.se>" or
"Harald Achitz <harald.achitz@getinge.com>", are copyright of Harald Achitz.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Harald Achitz
2017/03/21
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Anonymous Maarten
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "madebr", with
commit author "Anonymous Maarten <anonymous.maarten@gmail.com>", are copyright of Anonymous Maarten.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Anonymous Maarten
2017/03/27
# Permission to Relicense under MPLv2

This is a statement by Olaf Mandel
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "olafmandel", with
commit author "Olaf Mandel <o.mandel@menlosystems.com>", are copyright of Olaf Mandel.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Olaf Mandel
2017/03/21
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Juha Reunanen, Tomaattinen Ltd, and Outotec (Finland) Oy
that grants permission to relicense their copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the GitHub handle "reunanen", with
commit author "Juha Reunanen <juha.reunanen@tomaattinen.com>", are copyright of Juha Reunanen.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Juha Reunanen
2017/04/01

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Tom Whittock
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "twhittock", with
commit author "Tom Whittock", are copyright of Tom Whittock.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Tom Whittock
2017/04/10

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Caleb Epstein
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "Bklyn", with
commit author "Caleb Epstein <cae@bklyn.org>", are copyright of Caleb Epstein.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Caleb Epstein
2017/03/21

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Johan Mabille
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "JohanMabille", with
commit author "Johan Mabille johan.mabille@gmail.com", are copyright of Johan Mabille.

This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Johan Mabille
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Jovan Bunjevacki that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "bjovke", with commit author "Jovan Bunjevacki <bjovan@gmail.com>", are copyright of Jovan Bunjevacki. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Jovan Bunjevacki
2017/04/04

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Conrad Parker that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "kfish", with commit author "Conrad Parker <conrad@metadecks.org>", are copyright of Conrad Parker. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Conrad Parker
2018/10/30

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Timothee Besset
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "TTimo", with
commit author "Timothee Besset <ttimo@ttimo.net>", are copyright of Timothee Besset.
This document hereby grants the libzmq project team to relicense libzmq,

including all past, present and future contributions of the author listed above.

Timothee Besset
2017/03/19
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Francesco Montorsi that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "f18m", with commit author "Francesco Montorsi <francesco.montorsi@gmail.com>", are copyright of Francesco Montorsi. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Francesco Montorsi
2018/09/14
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Gennady Makovetski that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "gena-moscow", with commit author "Gennady Makovetski <makovetski@gmail.com>", are copyright of Gennady Makovetski. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Gennady Makovetski
2017/03/21
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Sbastien Rombauts that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "SRombauts", with commit author "Sbastien Rombauts", are copyright of Sbastien Rombauts. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Sbastien Rombauts

2017/03/21
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Pawel Kurdybacha
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "kurdybacha", with
commit author "Pawel Kurdybacha pawel.kurdybacha@gmail.com", are copyright of Pawel Kurdybacha .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Pawel Kurdybacha
2017/04/13
# Permission to Relicense under MPLV2

This document collects grants from firms that hold copyrights in ZeroMQ. Please add new firms at the start. Each patch must be made with a GitHub handle that guarantees identity of the signatory.

## iMatix Corporation

This is a statement by iMatix Corporation sprl (iMatix) that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

1. The original ZeroMQ codebase was developed from 2007-2009 by FastMQ Inc. (FastMQ) in Slovakia, which held all copyrights in code written by its staff.

2. On 2009-11-01, iMatix exercised an option to acquire FastMQ and its assets,
including ZeroMQ. On or before 2010-01-05 the FastMQ copyrights in ZeroMQ were
transferred to iMatix
([commit](https://github.com/zeromq/libzmq/commit/4f6baf4dde627656b63cc4e2acdb78a8577ba640)). The FastMQ legal entity was subsequently liquidated.

3. Between 2009 and 2016, iMatix has continued to add contributions to ZeroMQ under the GitHub handle 'hintjens'.

4. This grant therefore covers what remains of the original FastMQ codebase plus all later contributions.

5. iMatix hereby grants an irrevocable, global, and fully paid-up license on all its copyrights that exist in ZeroMQ, under the MPLv2.

-Pieter Hintjens
CEO, iMatix Corporation sprl
23 April 2016
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Simon Giesecke that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "sigiesec", with commit author "Simon Giesecke <simon.giesecke@btc-ag.com>", are copyright of Simon Giesecke. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Simon Giesecke
2017/08/17

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Erik Hugne that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "Hugne", with commit author "erik.hugne@gmail.com", are copyright of Erik Hugne.
Commits made by author "erik.hugne@ericsson.com" are NOT covered by this statement.
This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Erik Hugne
2017/03/22

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Alex Grnholm that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "agronholm", with commit author "Alex Grnholm <alex.gronholm+git@nextday.fi>", are copyright of Alex Grnholm.
This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Alex Grnholm
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement byEvgeny (Jim) Klimov
that grants permission to relicense his copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "jimklimov",
with commit author "Jim Klimov <jim@jimklimov.com>" or
"Jim Klimov <jimklimov@gmail.com>", are copyright of Evgeny (Jim) Klimov.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed
above.

Evgeny (Jim) Klimov
2017/03/21

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Vincent Tellier
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "vtellier", with
commit author "Vincent Tellier", are copyright of Vincent Tellier .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Vincent Louis Thophile Tellier
2017/03/20

## Brocade Communications Systems Inc.

This is a statement by Brocade Communications Systems Inc. (Brocade)
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "bluca", with
commit author "Luca Boccassi <luca.boccassi@gmail.com>" or
"Luca Boccassi <lboccass@brocade.com>", are copyright of Brocade.
This permission to relicense includes all past, present and future
contributions of Brocade employees.

Luca Boccassi
Software Engineer, Brocade Communications Systems Inc.
2016-05-16

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Andre Caron

that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "AndreLouisCaron", with
commit author "Andre Caron <andre.l.caron@gmail.com>", are copyright of Andre Caron.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Andre Caron
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Anton Dimitrov that grants permission to
relicense its copyrights in the libzmq C++ library (ZeroMQ) under the
Mozilla Public License v2 (MPLv2) or any other Open Source Initiative
approved license chosen by the current ZeroMQ BDFL (Benevolent
Dictator for Life).

A portion of the commits made by the Github handle "glemercier", with
commit author "Gregory Lemercier <greglemercier@free.fr>", are
copyright of Gregory Lmercier. This document hereby grants the libzmq
project team to relicense libzmq, including all past, present and
future contributions of the author listed above.

Gregory Lemercier

2018/10/07

# Permission to Relicense under MPLv2

This is a statement by Matthias Gabriel
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "gabm", with
commit author "Matthias Gabriel <matthias.gabriel@etit.tu-chemnitz.de>", are copyright of Matthias Gabriel.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Matthias Gabriel
2018/10/8

## Naos Ltd (a New Zealand company)

This is a statement by Naos Ltd (Naos) that grants permission to
relicense its copyrights in the libzmq C++ library (ZeroMQ) under
the Mozilla Public License v2 (MPLv2).

The port of libzmq to run on the z/OS Mainframe ([GitHub Pull
request #1136](https://github.com/zeromq/libzmq/pull/1136), [GitHub Pull
request #1138](https://github.com/zeromq/libzmq/pull/1138), and
[GitHub Pull request #1139](https://github.com/zeromq/libzmq/pull/1139))
was performed as work for hire under contract to iMatix Corporation
sprl, itself under contract to a client. Thus copyright in that
portability work does not belong to Naos Ltd, and Naos Ltd hereby
releases any claim to the copyright in the z/OS Mainframe portability
work identified by the above three GitHub pull requests.

Ewen McNeill
Managing Director, Naos Ltd
2016-04-25

# Permission to Relicense under MPLv2

This is a statement by Juraj Oruli that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "ojura", with commit author "Juraj Oruli", are copyright of Juraj Oruli.
This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Juraj Oruli
2018/10/26

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Henri Gourvest that grants permission to
relicense its copyrights in the libzmq C++ library (ZeroMQ) under the
Mozilla Public License v2 (MPLv2) or any other Open Source Initiative
approved license chosen by the current ZeroMQ BDFL (Benevolent
Dictator for Life).
A portion of the commits made by the Github handle "hgourvest", with
commit author "Henri Gourvest <hgourvest@progdigy.com>", are
copyright of Henri Gourvest. This document hereby grants the libzmq
project team to relicense libzmq, including all past, present and
future contributions of the author listed above.
Henri Gourvest
2018/12/8

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by G. Evan Burkitt
that grants permission to relicense his copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "eburkitt", with
commit author "Evan Burkitt <evanb@edulinksys.com>", are copyright of G. Evan Burkitt.

This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

G. Evan Burkitt
2017/04/06

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Szekely Gyorgy
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "hoditohod", with
commit author "Szekely Gyorgy <hoditohod@gmail.com>", are copyright of Szekely Gyorgy.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Szekely Gyorgy
2018/08/15

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Paul Colomiets
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "tailhook", with
commit author "Paul Colomiets <paul@colomiets.name>" or "Paul Colomiets
<pc@gafol.net>", are copyright of Paul Colomiets. This document hereby grants
the libzmq project team to relicense libzmq, including all past, present and
future contributions of the author listed above.

Paul Colomiets
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Mrio Kauba
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "soulik", with
commit author "Mrio Kauba mario.kasuba@it-academy.sk", are copyright of Mrio Kauba.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Mrio Kauba
2017/03/20
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Andrey Sibiryov that grants permission to relicense its copyrights in the
libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source
Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "kobolog", with commit author "Andrey Sibiryov
<me@kobology.ru>",
are copyright of Andrey Sibiryov. This document hereby grants the libzmq project team to
relicense libzmq, including all past, present and future contributions of the author listed above.

Andrey Sibiryov
2017/04/04
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Chuck Remes
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "chuckremes", with
commit author "Chuck Remes", are copyright of Chuck Remes.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Chuck Remes
2017/03/21
# Permission to Relicense under MPLv2

This is a statement by Jrmie Courrges-Anglas
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "jcourreges", with
commit author "Jrmie Courrges-Anglas", are copyright of Jrmie Courrges-Anglas.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Jrmie Courrges-Anglas
2018/12/07
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Uli Khler.
that grants permission to relicense its copyrights in the libzmq C++

library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

All commits made by the Github handle "ulikoehler", with
commit author "Uli Khler", are copyright of Uli Khler.

This document hereby grants the libzmq project team to relicense libzmq and / or CZMQ,
including all past, present and future contributions of the author listed above.

Uli Khler
2017/03/25
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Aaron Linville
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "linville", with
commit author "Aaron Linville <aaron@linville.org>", are copyright of Aaron Linville.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Aaron Linville
2017/03/27
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Rik van der Heijden
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "rikvdh", with
commit author "Rik van der Heijden <mail@rikvanderheijden.com>" and "Rik van der Heijden
<rik.vanderheijden@dualinventive.com>", are copyright of Rik van der Heijden.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Rik van der Heijden
2017/04/23
# Permission to Relicense under MPLv2

This is a statement by Chris Staite that grants permission to relicense its
copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License
v2 (MPLv2).

A portion of the commits made by the Github handle "chrisstaite", with commit
author "Chris <chris@yourdreamnet.co.uk>", are copyright of Chris Staite.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed
above.

Chris Staite
2017/03/28
# Permission to Relicense under MPLv2 or any other share-alike OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Andreas Rottmann that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other share-alike Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "rotty", with commit author "Andreas Rottmann <a.rottmann@gmx.at>", are copyright of Andreas Rottmann. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Andreas Rottmann
2017/05/07
# Permission to Relicense under MPLv2

This is a statement by KIU Shueng Chuan
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "pijyoi", with
commit author "KIU Shueng Chuan <nixchuan@gmail.com>", are copyright of KIU Shueng Chuan.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

KIU Shueng Chuan
2017/03/23
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Ghislain PUTOIS
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "ghpu", with

commit author "Ghislain PUTOIS", are copyright of Ghislain PUTOIS.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Ghislain PUTOIS
2017/03/20
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by crocket
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "crocket", with
commit author "crocket <748856+crocket@users.noreply.github.com>", are copyright of crocket.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

crocket
2018/11/10
Permission to Relicense under MPLv2 or any other OSI approved license
chosen by the current ZeroMQ BDFL

This is a statement by Stoian Ivanov that grants permission to relicense
its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public
License v2 (MPLv2) or any other Open Source Initiative approved license
chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "sdrsdr", with commit
author "Stoian Ivanov sdr@mail.bg", are copyright of Stoian Ivanov. This
document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Stoian Ivanov
2017-03-23
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Michael Lutz
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "michicc", with
commit author "Michael Lutz <michi@icosahedron.de>", are copyright of Michael Lutz.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Michael Lutz
2017/05/01
# Permission to Relicense under MPLv2

This is a statement by Kevin Sapper that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "sappo", with commit author "Kevin Sapper <mail@kevinsapper.de>", are copyright of Kevin Sapper. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Kevin Sapper
2017/03/20
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Matthew Connolly that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "mattconnolly", with commit author "Matt Connolly <matt.connolly@me.com>", are copyright of Matthew Connolly. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Matthew Connolly
2017/06/28
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Volodymyr Korniichuk that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "ezhikus", with commit author "VolodymyrKorn@gmail.com", are copyright of Volodymyr Korniichuk. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Volodymyr Korniichuk
2017/03/21
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by David Jelenc that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "djelenc", with commit author "David Jelenc <david.jelenc@fri.uni-lj.si>", are copyright of David Jelenc. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

David Jelenc
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Lionel Flandrin that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "simias", with commit author "Lionel Flandrin lflandrin@ereca.fr", are copyright of Lionel Flandrin. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Lionel Flandrin
2018/05/02

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Shubham Lagwankar that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "ssbl", with commit author "Shubham Lagwankar <shubhu105@gmail.com>", are copyright of Shubham Lagwankar. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Shubham Lagwankar
2018/10/30

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Joe Eli McIlvain that grants permission to relicense its copyrights in the libzmq C++

library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "jemc", with
commit author "Joe Eli McIlvain", are copyright of Joe Eli McIlvain.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Joe Eli McIlvain
2017/03/28

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Gian Lorenzo Meocci
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "meox", with
commit author "Gian Lorenzo Meocci <glmeocci@gmail.com>", are copyright of Gian Lorenzo Meocci.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Gian Lorenzo Meocci
2017/04/09

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Jan Kryl
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "jkryl", with
commit author "Jan Kryl jan.kryl@nexenta.com", are copyright of Jan Kryl.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Jan Kryl
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Michael Fox
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ

BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "gitfoxi", with
commit author "gitfoxi", are copyright of Michael Fox.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Michael Fox
2017/03/20

# Permission to Relicense under MPLv2 or any other share-alike OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Chia-liang Kao
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
share-alike Open Source Initiative approved license chosen by the current
ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "clkao", with
commit author "Chia-liang Kao clkao@clkao.org", are copyright of Chia-liang Kao.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Chia-liang Kao
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Christopher Dolan
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "cdolan", with
commit author "Christopher Dolan <chris@codingstream.org>", are copyright of Christopher Dolan.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Christopher Dolan
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Bill Torpey that grants permission to
relicense its copyrights in the libzmq C++ library (ZeroMQ) under the
Mozilla Public License v2 (MPLv2) or any other Open Source Initiative
approved license chosen by the current ZeroMQ BDFL (Benevolent
Dictator for Life).

A portion of the commits made by the Github handle "wallstprog", with
commit author "Bill Torpey <wallstprog@gmail.com>", are
copyright of Bill Torpey. This document hereby grants the libzmq
project team to relicense libzmq, including all past, present and
future contributions of the author listed above.

Bill Torpey
2018/11/02

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by H. Eduardo Montoya Snchez
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "montoyaedu", with
commit author "Montoya Edu <montoya.edu@gmail.com>", are copyright of H. Eduardo Montoya Snchez.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

H. Eduardo Montoya Snchez
2017/03/20

# Permission to Relicense under MPLv2

This is a statement by Roal Zanazzi that grants permission to relicense its
copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License
v2 (MPLv2).

A portion of the commits made by the Github handle "roalz", with commit author
"roalz <roal.zanazzi@gmail.com>", are copyright of Roal Zanazzi.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Roal Zanazzi
2017/03/21

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Yuval Langer
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "yuvallanger", with
commit author "Yuval Langer <yuval.langer@gmail.com>", are copyright of Yuval Langer.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Yuval Langer
2017-03-19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Simon Giesecke that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "sigiesec", with commit author "Simon Giesecke <simon.giesecke@btc-ag.com>", are copyright of Simon Giesecke. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Jeff Brubaker
2018/11/18

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Michael Hand that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "mipaaa", with commit author "Michael Hand <mipaaa@gmail.com>", are copyright of Michael Hand. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Michael Hand
2017/03/21

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Jacques Germishuys that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "jacquesg", with commit author "Jacques Germishuys <jacquesg@striata.com>", are copyright of Jacques Germishuys. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Jacques Germishuys
2018/12/14

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Wenbin Hou
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "Wenbin Hou", with
commit author "Wenbin Hou <houwenbin@pku.edu.cn>", are copyright of Wenbin Hou.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Wenbin Hou
2018/05/16

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by "chugga_fan"
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "chugga_fan", with
commit author "chugga_fan chuggafans@gmail.com", are copyright of "chugga_fan" .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

"chugga_fan"
2017/03/19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by William P Strang
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "willstrang", with
commit author "Will Strang <william.p.strang@gmail.com", are copyright of William P Strang.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

William P Strang
2017/03/23

# Permission to Relicense under MPLV2 or any Open Source Initiative (OSI) approved license chosen by the current ZeroMQ BDFL

This is a statement by Nick Guiffrida that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

The commits made by the Github handle "goodfella", with commit author "Nick Guiffrida goodfella005@gmail.com" are copyright of Nick Guiffrida. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Tore Halvorsen that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "demozon", with commit author "Tore Halvorsen <tore.halvorsen@gmail.com>", are copyright of Tore Halvorsen. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Tore Halvorsen
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Thomas Braun that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "t-b", with commit author "thomas.braun@virtuell-zuhause.de" and "thomas.braun@byte-physics.de", are copyright of Thomas Braun. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Thomas Braun
2017/03/27

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Christoph Schulz that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ

BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "Christoph Schulz", with
commit author "Christoph Schulz <schulzcbs@gmail.com>", are copyright of Christoph Schulz.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Christoph Schulz
2018/09/10

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Ming Ji
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "answeroo", with
commit author "answeroo <jiming@yafco.com>", are copyright of Ming Ji.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Ming Ji
2018/08/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Sergey KHripchenko
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "shripchenko", with
commit author "Sergey KHripchenko <shripchenko@intermedia.net>", are copyright of Sergey KHripchenko.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Sergey KHripchenko
2017/03/20

# Permission to Relicense under MPLv2

This is a statement by Antonio Vanegas that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "hpsaturn", with commit author "Antonio Vanegas", are copyright of Antonio Vanegas.
This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Antonio Vanegas

2018/10/27

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Franco Fichtner that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "fichtner", with commit author "Franco Fichtner <franco@lastsummer.de>", are copyright of Franco Fichtner. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Franco Fichtner

2017/03/21

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Ivo Danihelka that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "fidlej", with commit author "Ivo Danihelka <ivo@danihelka.net>", are copyright of Ivo Danihelka. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Ivo Danihelka

2017/03/21

# Permission to Relicense under MPLV2

This directory collects grants from individuals and firms that hold copyrights in ZeroMQ to permit licensing the ZeroMQ code under the [Mozilla Public License, version 2](https://www.mozilla.org/en-US/MPL/2.0/). See [GitHub Pull Request #1917](https://github.com/zeromq/libzmq/pull/1917), the [0MQ Licensing Page](http://zeromq.org/area:licensing) and [original iMatix zeromq-dev license grant](http://lists.zeromq.org/pipermail/zeromq-dev/2016-April/030288.html) for some background information.

Please create a separate file in this directory for each individual or firm holding copyright in ZeroMQ, named after the individual or firm holding the copyright.

Each patch must be made with a GitHub handle that is clearly
associated with the copyright owner, to guarantee the identity of
the signatory. Please avoid changing the files created by other
individuals or firms granting a copyright license over their
copyrights (if rewording is required contact them and ask them to
submit an updated version). This makes it easier to verify that
the license grant was made by an authorized GitHub account.
Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Lonard Michelet
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current
ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "leonarf", with
commit author "Andre Caron andre.l.caron@gmail.com", are copyright of Lonard Michelet.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Michelet Lonard 2017/03/29
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Martin Natano
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "natano", with
commit author "Martin Natano", are copyright of Martin Natano.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Martin Natano
2017/03/21
# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Diego R.Losada
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "drodri", with
commit author "drodri <diego.rlosada@gmail.com>", are copyright of Diego Rodriguez-Losada.
This document hereby grants the libzmq project team to relicense libzmq,

including all past, present and future contributions of the author listed above.

Diego Rodriguez-Losada
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Samuel Martin
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "tSed", with
commit author "Samuel Martin <s.martin49@gmail.com>", are copyright of Samuel Martin.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Samuel Martin
2017/03/25

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Byron Mallett
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "mystfit", with
commit author "Byron Mallett <byronated@gmail.com>", are copyright of Byron Mallett .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Byron Mallett
2018/9/1

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,

the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. 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 them 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 prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. 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.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey 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; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If 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 convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

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.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS 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.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by

the Free Software Foundation, either version 3 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, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program 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, your program's commands
might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

The GNU 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 Lesser General
Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

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 that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you

received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.

SPECIAL EXCEPTION GRANTED BY COPYRIGHT HOLDERS

As a special exception, copyright holders give you permission to link this
library with independent modules to produce an executable, regardless of
the license terms of these independent modules, and to copy and distribute
the resulting executable under terms of your choice, provided that you also
meet, for each linked independent module, the terms and conditions of
the license of that module. An independent module is a module which is not
derived from or based on this library. If you modify this library, you must
extend this exception to your version of the library.

Note: this exception relieves you of any obligations under sections 4 and 5
of this license, and section 6 of the GNU General Public License.
wepoll - epoll for Windows
https://github.com/piscisaureus/wepoll

Copyright 2012-2018, Bert Belder <bertbelder@gmail.com>
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.
The MIT License (MIT)

Copyright (c) <year> 2007-14 Mike Karlesky, Mark VanderVoord, Greg Williams

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.
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: ZeroMQ
Source: http://zeromq.org

Files: *
Copyright: 2009-2011, 250bpm s.r.o
2007-2013, iMatix Corporation
2007-2011, Other contributors as noted in the AUTHORS file
License: LGPL-3.0+

Files: src/xreq.cpp
src/xpub.cpp
src/xsub.cpp
tests/test_reqrep_device.cpp
tests/test_invalid_rep.cpp
Copyright: 2010-2011, 250bpm s.r.o
2011, VMware, Inc
2010-2011, Other contributors as noted in the AUTHORS file
License: LGPL-3.0+

Files: src/msg.hpp
src/xrep.*

src/options.*
src/req.*
src/socket_base.*
src/pipe.*
src/encoder.cpp
src/lb.cpp
src/session_base.*
src/fq.cpp
include/zmq.h
Copyright: 2009-2011, 250bpm s.r.o
2007-2013, iMatix Corporation
2011, VMware, Inc
2007-2011, Other contributors as noted in the AUTHORS file
License: LGPL-3.0+

Files: src/pgm_receiver.*
src/pgm_sender.*
src/pgm_socket.*
Copyright: 2009-2011, 250bpm s.r.o
2007-2013, iMatix Corporation
2010-2011, Miru Limited
2007-2011, Other contributors as noted in the AUTHORS file
License: LGPL-3.0+

Files: external/unity/*
Copyright: 2007-2014 Mike Karlesky
2007-2014 Mark VanderVoord
2007-2014 Greg Williams
License: MIT

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.

Files: debian/*

Copyright: 2014- , Laszlo Boszormenyi (GCS) <gcs@debian.org>
2012-2014, Alessandro Ghedini <ghedo@debian.org>
2010-2012, Martin Lucina <martin@lucina.net>
2009-2010, Adrian von Bidder <cmot@debian.org>
2009-2010, Peter Busser <peter@mirabilix.nl>
2012, Alessandro Ghedini <ghedo@debian.org>
License: LGPL-2.0+

License: LGPL-2.0+
This package 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 of the License, or (at your option) any later version.
.
This package 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 General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
.
On Debian systems, the complete text of the GNU Lesser General
Public License can be found in "/usr/share/common-licenses/LGPL-2".

License: LGPL-3.0+
This package 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 3 of the License, or (at your option) any later version.
.
This package 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 General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
.
On Debian systems, the complete text of the GNU Lesser General
Public License can be found in "/usr/share/common-licenses/LGPL-3".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.

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Adam Seering that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commit made by the Github handle "aseering", with commit author "Adam Seering <aseering@gmail.com>", are copyright of Adam Seering. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Adam Seering
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Christophe Juniet that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "cjuniet", with commit author "Christophe Juniet <c.juniet@gmail.com>", are copyright of Christophe Juniet. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Christophe Juniet
2017/03/20

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Guido Vranken that grants permission to relicense its copyrights in the libzmq C++ library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other Open Source Initiative approved license chosen by the current ZeroMQ BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "guidovranken", with commit author "Guido Vranken <guidovranken@gmail.com>", are copyright of Guido Vranken. This document hereby grants the libzmq project team to relicense libzmq, including all past, present and future contributions of the author listed above.

Guido Vranken
2019/01/08

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by John T. Conklin
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "johntconklin", with
commit author "J.T. Conklin <jtc@acorntoolworks.com>", are copyright of John T. Conklin.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

John T. Conklin
2017-03-19

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by Brian Buchanan
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ
BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "bwbuchanan", with
commit author "Brian Buchanan <bwb@holo.org>" or "Brian Buchanan <brian@tiogalake.com>", are copyright of Brian Buchanan.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Brian Buchanan
2017/03/23

# Permission to Relicense under MPLv2

This is a statement by Jake Cobb
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2).

A portion of the commits made by the Github handle "jakecobb", with
commit author "Jake Cobb <jake.cobb@gmail.com>", are copyright of Jake Cobb.
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

Jake Cobb
2017/04/04

# Permission to Relicense under MPLv2 or any other OSI approved license chosen by the current ZeroMQ BDFL

This is a statement by "chugga_fan"
that grants permission to relicense its copyrights in the libzmq C++
library (ZeroMQ) under the Mozilla Public License v2 (MPLv2) or any other
Open Source Initiative approved license chosen by the current ZeroMQ

BDFL (Benevolent Dictator for Life).

A portion of the commits made by the Github handle "chugga_fan", with
commit author "chugga_fan chuggafans@gmail.com", are copyright of "chugga_fan" .
This document hereby grants the libzmq project team to relicense libzmq,
including all past, present and future contributions of the author listed above.

"chugga_fan"
2017/03/19

1.61 lodash 4.17.10

1.61.1 Available under license :

Copyright JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as
documented below:

====

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 and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

Software License Agreement (BSD License)

Copyright (c) 2007, Parakey Inc.
All rights reserved.

Redistribution and use of this software 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 Parakey Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of Parakey Inc.

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.
Copyright (c) 2009-2016 Jeremy Ashkenas, DocumentCloud and Investigative
Reporters & Editors

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) 2010-2016 Jeremy Ashkenas, DocumentCloud

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 JS Foundation and other contributors <https://js.foundation/>

Based on Underscore.js, copyright Jeremy Ashkenas,
DocumentCloud and Investigative Reporters & Editors <http://underscorejs.org/>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/lodash/lodash

The following license applies to all parts of this software except as documented below:

====

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 and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====

Files located in the node_modules and vendor directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

1.62 logrus_fluent v0.5.1

1.62.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.63 luqum 0.7.5

1.63.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked

Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding

Source.)

5. Combined Libraries.

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 that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.64 mockito-core 1.10.19

1.65 moment 2.20.1

1.65.1 Available under license :

Copyright (c) JS Foundation 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.

1.66 netaddr 0.7.19

1.66.1 Available under license :

Here are the copyright notices applicable to the netaddr library.

----
netaddr

Copyright (c) 2008 by David P. D. Moss. All rights reserved.

Released under the BSD license. See the LICENSE file for details.

IANA (Internet Assigned Numbers Authority)
----

netaddr is not sponsored nor endorsed by IANA.

Use of data from IANA (Internet Assigned Numbers Authority) is subject to copyright and is provided with prior written permission.

IANA data files included with netaddr are not modified in any way but are parsed and made available to end users through an API.

See README file and source code for URLs to latest copies of the relevant files.

IEEE (Institution of Electrical Engineers)
----

netaddr is not sponsored nor endorsed by the IEEE.

Use of data from the IEEE (Institute of Electrical and Electronics Engineers) is subject to copyright. See the following URL for details :-

http://www.ieee.org/web/publications/rights/legal.html

IEEE data files included with netaddr are not modified in any way but are parsed and made available to end users through an API. There is no guarantee that referenced files are not out of date.

See README file and source code for URLs to latest copies of the relevant files.
=========
Copyright
=========

.. include:: ../../COPYRIGHT
============
Contributors
============

.. include:: ../../THANKS
=======
License
=======

.. include:: ../../LICENSE
Here are the licenses applicable to the use of the netaddr library.

----
netaddr

COPYRIGHT AND LICENSE

Copyright (c) 2008 by David P. D. Moss. 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 David P. D. Moss nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.67 numpy_Python 1.16.1

1.67.1 Available under license :

Copyright (c) 2005-2019, NumPy Developers.
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 NumPy Developers nor the names of any 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.

The NumPy repository and source distributions bundle several libraries that are compatibly licensed. We list these here.

Name: Numpydoc
Files: doc/sphinxext/numpydoc/*
License: 2-clause BSD
 For details, see doc/sphinxext/LICENSE.txt

Name: scipy-sphinx-theme
Files: doc/scipy-sphinx-theme/*
License: 3-clause BSD, PSF and Apache 2.0
 For details, see doc/scipy-sphinx-theme/LICENSE.txt

Name: lapack-lite
Files: numpy/linalg/lapack_lite/*
License: 3-clause BSD
 For details, see numpy/linalg/lapack_lite/LICENSE.txt

Name: tempita
Files: tools/npy_tempita/*
License: BSD derived
 For details, see tools/npy_tempita/license.txt

Name: dragon4
Files: numpy/core/src/multiarray/dragon4.c
License: One of a kind

For license text, see numpy/core/src/multiarray/dragon4.c
License

Copyright (c) 2008 Ian Bicking and 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.

****
NumPy License
****

Copyright (c) 2005, NumPy Developers

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 NumPy Developers nor the names of any
 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.
Copyright (C) 2008 Stefan van der Walt <stefan@mentat.za.net>, Pauli Virtanen <pav@iki.fi>

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 AUTHOR ``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 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 following copyright statement applies to all files, excluding exceptions mentioned explicitly below:

Copyright (c) 2011 Kevin Dunn, Surya K, Pauli Virtanen, the Sphinx team
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 author nor the names of other 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.

Exceptions:

_theme/scipy/static/copybutton.js

Originates from Python; Doc/tools/sphinxext/static/copybutton.js

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, 2013 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.

_theme/scipy/static/less/bootstrap/*,
_theme/scipy/static/img/glyphicons-*,
_theme/scipy/static/css/spc-bootstrap.css

Twitter Bootstrap v2.3.1 (and generated output CSS files)

Copyright 2012 Twitter, Inc
Licensed under the Apache License v2.0
http://www.apache.org/licenses/LICENSE-2.0
Copyright (c) 2005-2019, NumPy Developers.
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 NumPy Developers nor the names of any
  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.

The NumPy repository and source distributions bundle several libraries that are compatibly licensed. We list these here.

Name: Numpydoc
Files: doc/sphinxext/numpydoc/*
License: 2-clause BSD
 For details, see doc/sphinxext/LICENSE.txt

Name: scipy-sphinx-theme
Files: doc/scipy-sphinx-theme/*
License: 3-clause BSD, PSF and Apache 2.0
 For details, see doc/scipy-sphinx-theme/LICENSE.txt

Name: lapack-lite
Files: numpy/linalg/lapack_lite/*
License: 3-clause BSD
 For details, see numpy/linalg/lapack_lite/LICENSE.txt

Name: tempita
Files: tools/npy_tempita/*
License: BSD derived
 For details, see tools/npy_tempita/license.txt

Name: dragon4
Files: numpy/core/src/multiarray/dragon4.c
License: One of a kind
 For license text, see numpy/core/src/multiarray/dragon4.c
The author(s) grant permission for redistribution and use in source and
binary forms, with or without modification, of the software and documentation
provided that the following conditions are met:

0. If you receive a version of the software that is specifically labelled as not being for redistribution (check the version message and/or README), you are not permitted to redistribute that version of the software in any way or form.
1. All terms of all other applicable copyrights and licenses must be followed.
2. Redistributions of source code must retain the authors' copyright notice(s), this list of conditions, and the following disclaimer.
3. Redistributions in binary form must reproduce the authors' copyright notice(s), this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
4. All advertising materials mentioning features or use of this software must display the following acknowledgement with the name(s) of the authors as specified in the copyright notice(s) substituted where indicated:

This product includes software developed by the authors which are mentioned at the start of the source files and other contributors.

5. Neither the name(s) of the author(s) 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 ITS AUTHORS 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 AUTHORS 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.

1.68 object-assign 4.1.1

1.68.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.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.

1.69 object-assign 4.1.1

1.69.1 Available under license :

The MIT License (MIT)

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.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.

1.70 olivere-elastic 6.2.16

1.70.1 Available under license :

Copyright (c) 2013 Joshua Tacoma

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 (MIT)
Copyright 2012-2015 Oliver Eilhard

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.
# This is a list of people who have contributed code
# to the Elastic repository.
#
# It is just my small "thank you" to all those that helped
# making Elastic what it is.
#
# Please keep this list sorted.

0x6875790d0a [@huydx](https://github.com/huydx)
Adam Alix [@adamalix](https://github.com/adamalix)
Adam Weiner [@adamweiner](https://github.com/adamweiner)
Adrian Lungu [@AdrianLungu](https://github.com/AdrianLungu)
alehano [@alehano](https://github.com/alehano)
Alex [@akotlar](https://github.com/akotlar)
Alexander Sack [@asac](https://github.com/asac)

Alexandre Olivier [@aliphen](https://github.com/aliphen)
Alexey Sharov [@nizsheanez](https://github.com/nizsheanez)
AndreKR [@AndreKR](https://github.com/AndreKR)
Andr Bierlein [@ligustah](https://github.com/ligustah)
Andrew Dunham [@andrew-d](https://github.com/andrew-d)
Andrew Gaul [@andrewgaul](https://github.com/andrewgaul)
Andy Walker [@alaska](https://github.com/alaska)
Arquivei [@arquivei](https://github.com/arquivei)
arthurgustin [@arthurgustin](https://github.com/arthurgustin)
Benjamin Fernandes [@LotharSee](https://github.com/LotharSee)
Benjamin Zarzycki [@kf6nux](https://github.com/kf6nux)
Boris Popovschi [@Zyqsempai](https://github.com/Zyqsempai)
Braden Bassingthwaite [@bbassingthwaite-va](https://github.com/bbassingthwaite-va)
Brady Love [@bradylove](https://github.com/bradylove)
Bryan Conklin [@bmconklin](https://github.com/bmconklin)
Bruce Zhou [@brucez-isell](https://github.com/brucez-isell)
Carl Dunham [@carldunham](https://github.com/carldunham)
Cat [@cat-turner](https://github.com/cat-turner)
Csar Jimnez [@cesarjimenez](https://github.com/cesarjimenez)
cforbes [@cforbes](https://github.com/cforbes)
(Chang Tai Wei) [@david30907d](https://github.com/david30907d)
Chris M [@tebriel](https://github.com/tebriel)
Chris Rice [@donutmonger](https://github.com/donutmonger)
Claudiu Olteanu [@claudiuolteanu](https://github.com/claudiuolteanu)
Chris Duncan [@veqryn](https://github.com/veqryn)
Chris Ludden [@cludden](https://github.com/cludden)
Christophe Courtaut [@kri5](https://github.com/kri5)
cmitchell [@cmitchell](https://github.com/cmitchell)
Connor Peet [@connor4312](https://github.com/connor4312)
Conrad Pankoff [@deoxxa](https://github.com/deoxxa)
Corey Scott [@corsc](https://github.com/corsc)
Daniel Barrett [@shendaras](https://github.com/shendaras)
Daniel Heckrath [@DanielHeckrath](https://github.com/DanielHeckrath)
Daniel Imfeld [@dimfeld](https://github.com/dimfeld)
Daniel Santos [@danlsgiga](https://github.com/danlsgiga)
David Emanuel Buchmann [@wuurrd](https://github.com/wuurrd)
Dwayne Schultz [@myshkin5](https://github.com/myshkin5)
Ellison Leo [@ellisonleao](https://github.com/ellisonleao)
Erik Grinaker [@erikgrinaker](https://github.com/erikgrinaker)
Erwin [@eticzon](https://github.com/eticzon)
Etienne Lafarge [@elafarge](https://github.com/elafarge)
Eugene Egorov [@EugeneEgorov](https://github.com/EugeneEgorov)
Evan Shaw [@edsrzf](https://github.com/edsrzf)
Fanfan [@wenpos](https://github.com/wenpos)
Faolan C-P [@fcheslack](https://github.com/fcheslack)
Filip Tepper [@filiptepper](https://github.com/filiptepper)
Garrett Kelley [@GarrettKelley](https://github.com/GarrettKelley)
Gaspard Douady [@plopik](https://github.com/plopik)

Gaylord Aulke [@blafasel42](https://github.com/blafasel42)
Gerhard Hring [@ghaering](https://github.com/ghaering)
Guilherme Silveira [@guilherme-santos](https://github.com/guilherme-santos)
Guillaume J. Charmes [@creack](https://github.com/creack)
Guiseppe [@gm42](https://github.com/gm42)
Han Yu [@MoonighT](https://github.com/MoonighT)
Harmen [@alicebob](https://github.com/alicebob)
Harrison Wright [@wright8191](https://github.com/wright8191)
Henry Clifford [@hcliff](https://github.com/hcliff)
Igor Dubinskiy [@idubinskiy](https://github.com/idubinskiy)
initialcontext [@initialcontext](https://github.com/initialcontext)
Isaac Saldana [@isaldana](https://github.com/isaldana)
J Barkey Wolf [@jjhbw](https://github.com/jjhbw)
Jack Lindamood [@cep21](https://github.com/cep21)
Jacob [@jdelgad](https://github.com/jdelgad)
Jayme Rotsaert [@jrots](https://github.com/jrots)
Jean-Alexandre Beaumont [@Enteris](https://github.com/Enteris)
Jeff Rand [@jeffrand](https://github.com/jeffrand)
Jeremy Canady [@jrmycanady](https://github.com/jrmycanady)
Jrmie Vexiau [@texvex](https://github.com/texvex)
Jim Berlage [@jimberlage](https://github.com/jimberlage)
Joe Buck [@four2five](https://github.com/four2five)
John Barker [@j16r](https://github.com/j16r)
John Goodall [@jgoodall](https://github.com/jgoodall)
John Stanford [@jxstanford](https://github.com/jxstanford)
Jonas Groenaas Drange [@semafor](https://github.com/semafor)
Josef Frhle [@Dexus](https://github.com/Dexus)
Jos Martnez [@xose](https://github.com/xose)
Josh Chorlton [@jchorl](https://github.com/jchorl)
Jpnock [@Jpnock](https://github.com/Jpnock)
jun [@coseyo](https://github.com/coseyo)
Junpei Tsuji [@jun06t](https://github.com/jun06t)
kartlee [@kartlee](https://github.com/kartlee)
Keith Hatton [@khatton-ft](https://github.com/khatton-ft)
kel [@liketic](https://github.com/liketic)
Kenta SUZUKI [@suzuken](https://github.com/suzuken)
Kevin Mulvey [@kmulvey](https://github.com/kmulvey)
Kyle Brandt [@kylebrandt](https://github.com/kylebrandt)
Larry Cinnabar [@larrycinnabar](https://github.com/larrycinnabar)
Leandro Piccilli [@lpic10](https://github.com/lpic10)
Lee [@leezhm](https://github.com/leezhm)
lechnertech [@lechnertech](https://github.com/lechnertech)
M. Zulfa Achsani [@misterciput](https://github.com/misterciput)
Maciej Lisiewski [@c2h5oh](https://github.com/c2h5oh)
Mara Kim [@autochthe](https://github.com/autochthe)
Marcy Buccellato [@marcybuccellato](https://github.com/marcybuccellato)
Mark Costello [@mcos](https://github.com/mcos)
Martin Hger [@protomouse](https://github.com/protomouse)

Medhi Bechina [@mdzor](https://github.com/mdzor)
Mike Beshai [@mbesh](https://github.com/mbesh)
mmfrb [@mmfrb](https://github.com/mmfrb)
mnpritula [@mnpritula](https://github.com/mnpritula)
mosa [@mosasiru](https://github.com/mosasiru)
Muhammet akr [@cakirmuha](https://github.com/cakirmuha)
naimulhaider [@naimulhaider](https://github.com/naimulhaider)
Naoya Yoshizawa [@azihsoyn](https://github.com/azihsoyn)
navins [@ishare](https://github.com/ishare)
Naoya Tsutsumi [@tutuming](https://github.com/tutuming)
NeoCN [@NeoCN](https://github.com/NeoCN)
Nicholas Wolff [@nwolff](https://github.com/nwolff)
Nick K [@utrack](https://github.com/utrack)
Nick Whyte [@nickw444](https://github.com/nickw444)
Nicolae Vartolomei [@nvartolomei](https://github.com/nvartolomei)
Orne Brocaar [@brocaar](https://github.com/brocaar)
Paul [@eyeamera](https://github.com/eyeamera)
Paul Oldenburg [@lr-paul](https://github.com/lr-paul)
Pete C [@peteclark-ft](https://github.com/peteclark-ft)
Peter Nagy [@nagypeterjob](https://github.com/nagypeterjob)
Paolo [@ppiccolo](https://github.com/ppiccolo)
Radoslaw Wesolowski [r--w](https://github.com/r--w)
rchicoli [@rchicoli](https://github.com/rchicoli)
Roman Colohanin [@zuzmic](https://github.com/zuzmic)
Ryan Schmukler [@rschmukler](https://github.com/rschmukler)
Ryan Wynn [@rwynn](https://github.com/rwynn)
Sacheendra talluri [@sacheendra](https://github.com/sacheendra)
Sean DuBois [@Sean-Der](https://github.com/Sean-Der)
Sagan Yaroslav [@sgnrslv](https://github.com/sgnrslv)
Shalin LK [@shalinlk](https://github.com/shalinlk)
singham [@zhaochenxiao90](https://github.com/zhaochenxiao90)
Slawomir CALUCH [@slawo](https://github.com/slawo)
Stephan Krynauw [@skrynauw](https://github.com/skrynauw)
Stephen Kubovic [@stephenkubovic](https://github.com/stephenkubovic)
Stuart Warren [@Woz](https://github.com/stuart-warren)
Sulaiman [@salajlan](https://github.com/salajlan)
Sundar [@sundarv85](https://github.com/sundarv85)
Swarlston [@Swarlston](https://github.com/Swarlston)
Take [ww24](https://github.com/ww24)
Tetsuya Morimoto [@t2y](https://github.com/t2y)
TheZeroSlave [@TheZeroSlave](https://github.com/TheZeroSlave)
TimeEmit [@TimeEmit](https://github.com/timeemit)
TusharM [@tusharm](https://github.com/tusharm)
wangtuo [@wangtuo](https://github.com/wangtuo)
Wdney Yuri [@wedneyyuri](https://github.com/wedneyyuri)
Wesley Kim [@wesleyk](https://github.com/wesleyk)
wolfkdy [@wolfkdy](https://github.com/wolfkdy)
Wyndham Blanton [@wyndhblb](https://github.com/wyndhblb)

Yarden Bar [@ayashjorden](https://github.com/ayashjorden)
zakthomas [@zakthomas](https://github.com/zakthomas)
Yuya Kusakabe [@higebu](https://github.com/higebu)
Zach [@snowzach](https://github.com/snowzach)
zhangxin [@visaxin](https://github.com/visaxin)
@ [@zplzpl](https://github.com/zplzpl)
The MIT License (MIT)
Copyright © 2012-2015 Oliver Eilhard

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.

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.

1.71 org.apache.kafka/kafka-streams 2.1.1

1.71.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This distribution has a binary dependency on jersey, which is available under the CDDL
License as described below.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL - Version 1.1)

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.
2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.
3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.
4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.
COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE

OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.
6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.
UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.
The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.
This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.
As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

This distribution has a binary dependency on zstd, which is available under the BSD 3-Clause License as described below.

BSD License

For Zstandard software

Copyright (c) 2016-present, Facebook, 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 Facebook 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.

This distribution has a binary dependency on zstd-jni, which is available under the BSD 2-Clause License
as described below.

Zstd-jni: JNI bindings to Zstd Library

Copyright (c) 2015-2016, Luben Karavelov/ All rights reserved.

BSD License

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.
Kafka Streams Scala
Copyright (C) 2018 Lightbend Inc. <https://www.lightbend.com>

Copyright (C) 2017-2018 Alexis Seigneurin.
Apache Kafka
Copyright 2019 The Apache Software Foundation.

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This distribution has a binary dependency on jersey, which is available under the CDDL
License. The source code of jersey can be found at https://github.com/jersey/jersey/.

1.72 org.springframework.boot 2.0.1.RELEASE

1.72.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Copyright 2012-2013 the original author or authors.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
open_source_licenses.txt

Spring Boot CLI
===

Pivotal makes available all content in this download ("Content").
Unless otherwise indicated below, the Content is provided to you under
the terms and conditions of the Apache License 2.0 (the "License"). A

copy of the license is available in the file called LICENSE.txt or you
may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

The following copyright statements and licenses apply to various open source software packages (or portions thereof) that are distributed with this content.

==

TABLE OF CONTENTS

==

The following is a listing of the open source components detailed in this document. This list is provided for your convenience; please read further if you wish to review the copyright notice(s) and the full text of the license associated with each component.

SECTION 1: BSD-STYLE, MIT-STYLE, OR SIMILAR STYLE LICENSES

SECTION 2: Apache License, V2.0

SECTION 3: Eclipse Public License, Version 1.0

>>> Aether API (org.eclipse.aether:aether-api)
>>> Aether Connector Basic (org.eclipse.aether:aether-connector-basic)
>>> Aether Implementation (org.eclipse.aether:aether-impl)
>>> Aether SPI (org.eclipse.aether:aether-spi)
>>> Aether Transport File (org.eclipse.aether:aether-transport-file)
>>> Aether Transport HTTP (org.eclipse.aether:aether-transport-http)
>>> Aether Utilities (org.eclipse.aether:aether-util)

--------------- SECTION 1: BSD-STYLE, MIT-STYLE, OR SIMILAR STYLE LICENSES ----------

BSD-STYLE, MIT-STYLE, OR SIMILAR STYLE LICENSES are applicable to the following component(s).

>>> JLine (jline:jline)

Copyright (c) 2002-2006, Marc Prud'hommeaux <mwp1@cornell.edu>
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 JLine 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.

>>> net.sf.jopt-simple:jopt-simple:4.5

The MIT License (MIT)

Copyright (c) <year> <copyright holders>

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.

>>> org.ow2.asm:asm

Copyright (c) 2000-2011 INRIA, France Telecom
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. Neither the name of the copyright holders 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.

-------------- SECTION 2: Apache License, V2.0 ----------

Apache License, V2.0 is applicable to the following component(s).

>>> org.apache.httpcomponents:httpclient
>>> org.apache.httpcomponents:httpcore
>>> org.sonatype.plexus:plexus-cipher
>>> org.sonatype.plexus:plexus-sec-dispatcher
>>> commons-logging:commons-logging
>>> org.codehaus.groovy:groovy
>>> org.apache.maven:maven-aether-provider
>>> org.apache.maven:maven-model
>>> org.apache.maven:maven-model-builder
>>> org.apache.maven:maven-repository-metadata
>>> org.apache.maven:maven-settings
>>> org.apache.maven:maven-settings-builder
>>> org.codehaus.plexus:plexus-component-annotations
>>> org.codehaus.plexus:plexus-utils
>>> org.codehaus.plexus:plexus-component-api
>>> org.codehaus.plexus:plexus-interpolation

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

>>> CGLIB 3.0 (cglib:cglib:3.0):

Per the LICENSE file in the CGLIB JAR distribution downloaded from
http://sourceforge.net/projects/cglib/files/cglib3/3.0/cglib-3.0.jar/download,
CGLIB 3.0 is licensed under the Apache License, version 2.0, the text of which
is included above.

--------------- SECTION 3: Eclipse Public License, Version 1.0 ----------

Eclipse Public License, Version 1.0 is applicable to the following component(s).

>>> org.eclipse.aether:aether-api
>>> org.eclipse.aether:aether-connector-basic
>>> org.eclipse.aether:aether-impl
>>> org.eclipse.aether:aether-spi
>>> org.eclipse.aether:aether-transport-file
>>> org.eclipse.aether:aether-transport-http
>>> org.eclipse.aether:aether-util

The Eclipse Foundation makes available all content in this plug-in ("Content"). Unless otherwise indicated below, the Content is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is available at http://www.eclipse.org/legal/epl-v10.html.

For purposes of the EPL, "Program" will mean the Content.

If you did not receive this Content directly from the Eclipse Foundation, the Content is being redistributed by another party ("Redistributor") and different terms and conditions may apply to your use of any object code in the Content. Check the Redistributor's license that was provided with the Content. If no such license exists, contact the Redistributor. Unless otherwise indicated below, the terms and conditions of the EPL still apply to any source code in the Content and such source code may be obtained at http://www.eclipse.org/

===

To the extent any open source subcomponents are licensed under the EPL and/or other similar licenses that require the source code and/or modifications to source code to be made available (as would be noted above), you may obtain a copy of the source code corresponding to the binaries for such open source components and modifications thereto, if any, (the "Source Files"), by downloading the Source Files from https://github.com/spring-projects/spring-boot, or by sending a request, with your name and address to:

Pivotal, Inc., 875 Howard St,
San Francisco, CA 94103
United States of America

or email info@pivotal.io. All such requests should clearly specify:

OPEN SOURCE FILES REQUEST
Attention General Counsel

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a

copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following

boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.73 org.springframework.kafka/spring-kafka 2.1.9.RELEASE

1.73.1 Available under license :

Spring for Apache Kafka Framework ${version}
Copyright (c) 2016-${copyright} Pivotal, Inc.

This product is licensed to you under the Apache License, Version 2.0
(the "License"). You may not use this product except in compliance with
the License.

This product may include a number of subcomponents with separate
copyright notices and license terms. Your use of the source code for
these subcomponents is subject to the terms and conditions of the
subcomponent's license, as noted in the license.txt file.

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 testData1) 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 testData1 and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [testData1 of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.

1.74 perks v1.0.0

1.74.1 Available under license :

Copyright (C) 2013 Blake Mizerany

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.

1.75 pierrec/lz4 v2.2.0

1.75.1 Available under license :

Copyright (c) 2015, Pierre Curto
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 xxHash 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.

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. 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 them 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 prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. 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.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological

measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

4. Conveying Verbatim Copies.

You may convey 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;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.

b) The work must carry prominent notices stating that it is
released under this License and any conditions added under section
7. This requirement modifies the requirement in section 4 to
"keep intact all notices".

c) You must license the entire work, as a whole, under this
License to anyone who comes into possession of a copy. This
License will therefore apply, along with any applicable section 7
additional terms, to the whole of the work, and all its parts,
regardless of how they are packaged. This License gives no
permission to license the work in any other way, but it does not
invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If 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 convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU 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 that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

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.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS 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.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 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, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
This program 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, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU 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 Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

1.76 pierrec/xxHash v0.1.5

1.76.1 Available under license :

Copyright (c) 2014, Pierre Curto
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 xxHash 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.

1.77 pkg-errors v0.9

1.77.1 Available under license :

Copyright (c) 2015, Dave Cheney <dave@cheney.net>
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 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.

1.78 prometheus v2.5.0

1.78.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Common libraries shared by Prometheus Go components.
Copyright 2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Data model artifacts for Prometheus.
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).
procfs provides functions to retrieve system, kernel and process
metrics from the pseudo-filesystem proc.

Copyright 2014-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).
Prometheus instrumentation library for Go applications
Copyright 2012-2015 The Prometheus Authors

This product includes software developed at
SoundCloud Ltd. (http://soundcloud.com/).

The following components are included in this product:

perks - a fork of https://github.com/bmizerany/perks
https://github.com/beorn7/perks
Copyright 2013-2015 Blake Mizerany, Bjrn Rabenstein
See https://github.com/beorn7/perks/blob/master/README.md for license details.

Go support for Protocol Buffers - Google's data interchange format
http://github.com/golang/protobuf/
Copyright 2010 The Go Authors
See source code for license details.

Support for streaming Protocol Buffer messages for the Go language (golang).
https://github.com/matttproud/golang_protobuf_extensions
Copyright 2013 Matt T. Proud

Licensed under the Apache License, Version 2.0

1.79 prop-types 15.6.1

1.79.1 Available under license :

MIT License

Copyright (c) 2013-present, Facebook, Inc.

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.

1.80 prop-types 15.6.1

1.80.1 Available under license :

MIT License

Copyright (c) 2013-present, Facebook, Inc.

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

Copyright (c) 2013-present, Facebook, Inc.

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.
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.

1.81 protobuf java 3.2.0

1.81.1 Available under 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>
This file contains a list of people who have made large contributions
to the public version of Protocol Buffers.

Original Protocol Buffers design and implementation:

Sanjay Ghemawat <sanjay@google.com>
Jeff Dean <jeff@google.com>
Daniel Dulitz <daniel@google.com>
Craig Silverstein
Paul Haahr <haahr@google.com>
Corey Anderson <corin@google.com>
(and many others)

Proto2 C++ and Java primary author:
Kenton Varda <kenton@google.com>

Proto2 Python primary authors:
Will Robinson <robinson@google.com>
Petar Petrov <petar@google.com>

Java Nano primary authors:
Brian Duff <bduff@google.com>
Tom Chao <chaot@google.com>
Max Cai <maxtroy@google.com>
Ulas Kirazci <ulas@google.com>

Large code contributions:
Jason Hsueh <jasonh@google.com>
Joseph Schorr <jschorr@google.com>
Wenbo Zhu <wenboz@google.com>

Large quantity of code reviews:
Scott Bruce <sbruce@google.com>
Frank Yellin
Neal Norwitz <nnorwitz@google.com>
Jeffrey Yasskin <jyasskin@google.com>
Ambrose Feinstein <ambrose@google.com>

Documentation:
Lisa Carey <lcarey@google.com>

Maven packaging:
Gregory Kick <gak@google.com>

Patch contributors:
Kevin Ko <kevin.s.ko@gmail.com>
  * Small patch to handle trailing slashes in --proto_path flag.
Johan Euphrosine <proppy@aminche.com>
  * Small patch to fix Python CallMethod().
Ulrich Kunitz <kune@deine-taler.de>
  * Small optimizations to Python serialization.
Leandro Lucarella <llucax@gmail.com>
  * VI syntax highlighting tweaks.

  * Fix compiler to not make output executable.
Dilip Joseph <dilip.antony.joseph@gmail.com>
  * Heuristic detection of sub-messages when printing unknown fields in
    text format.
Brian Atkinson <nairb774@gmail.com>
  * Added @Override annotation to generated Java code where appropriate.
Vincent Choinire <Choiniere.Vincent@hydro.qc.ca>
  * Tru64 support.
Monty Taylor <monty.taylor@gmail.com>
  * Solaris 10 + Sun Studio fixes.
Alek Storm <alek.storm@gmail.com>
  * Slicing support for repeated scalar fields for the Python API.
Oleg Smolsky <oleg.smolsky@gmail.com>
  * MS Visual Studio error format option.
  * Detect unordered_map in stl_hash.m4.
Brian Olson <brianolson@google.com>
  * gzip/zlib I/O support.
Michael Poole <mdpoole@troilus.org>
  * Fixed warnings about generated constructors not explicitly initializing
    all fields (only present with certain compiler settings).
  * Added generation of field number constants.
Wink Saville <wink@google.com>
  * Fixed initialization ordering problem in logging code.
Will Pierce <willp@nuclei.com>
  * Small patch improving performance of in Python serialization.
Alexandre Vassalotti <alexandre@peadrop.com>
  * Emacs mode for Protocol Buffers (editors/protobuf-mode.el).
Scott Stafford <scott.stafford@gmail.com>
  * Added Swap(), SwapElements(), and RemoveLast() to Reflection interface.
Alexander Melnikov <alm@sibmail.ru>
  * HPUX support.
Oliver Jowett <oliver.jowett@gmail.com>
  * Detect whether zlib is new enough in configure script.
  * Fixes for Solaris 10 32/64-bit confusion.
Evan Jones <evanj@mit.edu>
  * Optimize Java serialization code when writing a small message to a stream.
  * Optimize Java serialization of strings so that UTF-8 encoding happens only
    once per string per serialization call.
  * Clean up some Java warnings.
  * Fix bug with permanent callbacks that delete themselves when run.
Michael Kucharski <m.kucharski@gmail.com>
  * Added CodedInputStream.getTotalBytesRead().
Kacper Kowalik <xarthisius.kk@gmail.com>
  * Fixed m4/acx_pthread.m4 problem for some Linux distributions.
William Orr <will@worrbase.com>
  * Fixed detection of sched_yield on Solaris.
  * Added atomicops for Solaris
Andrew Paprocki <andrew@ishiboo.com>

* Fixed minor IBM xlC compiler build issues
* Added atomicops for AIX (POWER)

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.
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.
# 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>

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.

1.82 protobuf-1.2.0 1.2.0

1.82.1 Available under license :

Copyright 2010 The Go 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 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.

1.83 protobuf-java-format 1.4

1.83.1 Available under license :

Copyright (c) 2009, Orbitz World Wide
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 Orbitz World Wide nor the names of its contributors
   may be used to endorse or promote products derived from this software
   without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.84 proxy-polyfill 0.3.0

1.84.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.85 query-string 6.1.0

1.85.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.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.

1.86 query-string 5.1.1

1.86.1 Available under license :

MIT License

Copyright (c) Sindre Sorhus <sindresorhus@gmail.com> (sindresorhus.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.

1.87 rc-calendar 9.6.2

1.87.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-present yiminghe

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.

1.88 rc-time-picker 3.3.1

1.88.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-present yiminghe

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.

1.89 rcrowley/go-metrics master

1.89.1 Available under license :

Copyright 2012 Richard Crowley. 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.

THIS SOFTWARE IS PROVIDED BY RICHARD CROWLEY ``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 RICHARD CROWLEY 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 Richard Crowley.

1.90 react 16.3.2

1.90.1 Available under license :

MIT License

Copyright (c) 2013-present, Facebook, Inc.

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.

1.91 react-addons-shallow-compare 15.6.2

1.91.1 Available under license :

MIT License

Copyright (c) 2013-present, Facebook, Inc.

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.

1.92 react-dates 12.7.1

1.92.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Airbnb

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 (MIT)

Copyright (c) 2016 Airbnb

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.
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.

1.93 react-datetime 2.14.0

1.93.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017 Javier Marquez

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.

1.94 react-dom 16.3.2

1.94.1 Available under license :

MIT License

Copyright (c) 2013-present, Facebook, Inc.

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

Copyright (c) 2013-present, Facebook, Inc.

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.

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.

1.95 react-form 3.5.5

1.95.1 Available under license :

MIT License

Copyright (c) 2016 Tanner Linsley

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

Copyright (c) 2016 Tanner Linsley

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.
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.

1.96 react-onclickoutside 6.7.1

1.97 react-popper 0.10.1

1.97.1 Available under license :

The MIT License (MIT)

Copyright (c) 2018 React Popper authors

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.

1.98 react-redux 5.0.7

1.98.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

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.

1.99 react-router-dom 4.2.2

1.100 react-router-redux 5.0.0

1.100.1 Available under license :

Copyright (c) 2015 James Long

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.

1.101 react-router-redux 5.0.0-alpha.9

1.101.1 Available under license :

Copyright (c) 2015-present James Long

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.

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.

1.102 react-select 1.2.1

1.102.1 Available under license :

The MIT License (MIT)

Copyright (c) 2018 Jed Watson

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.

1.103 redux 4.0.0

1.103.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

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.

1.104 redux-promise-middleware 5.1.1

1.104.1 Available under license :

# Contributors

- [pburtchaell](https://github.com/pburtchaell)
- [tomatau](https://github.com/tomatau)
- [mik01aj](https://github.com/mik01aj)
- [joshkel](https://github.com/joshkel)
- [ianks](https://github.com/ianks)

- [therewillbecode](https://github.com/therewillbecode)
- [doronpr](https://github.com/doronpr)
- [sandiiarov](https://github.com/sandiiarov)
- [andrgood](https://github.com/andrgood)
- [mironov](https://github.com/mironov)
- [bennypowers](https://github.com/bennypowers)
- [cedricmessiant](https://github.com/cedricmessiant)
- [danwang](https://github.com/danwang)
- [GeKorm](https://github.com/GeKorm)
- [notgiorgi](https://github.com/notgiorgi)
- [jarsbe](https://github.com/jarsbe)
- [janeklb](https://github.com/janeklb)
- [jmar777](https://github.com/jmar777)
- [kishorevarma](https://github.com/kishorevarma)
- [kuuup](https://github.com/kuuup)
- [pzenger](https://github.com/pzenger)
- [pjvds](https://github.com/pjvds)
- [piu130](https://github.com/piu130)
- [randycoulman](https://github.com/randycoulman)
- [rotemmiz](https://github.com/rotemmiz)
- [toddmazierski](https://github.com/toddmazierski)
- [umidbekkarimov](https://github.com/umidbekkarimov)
- [anajavi](https://github.com/anajavi)
- [comerc](https://github.com/comerc)
- [extg](https://github.com/extg)
- [mikew](https://github.com/mikew)
- [roboslone](https://github.com/roboslone)

The MIT License (MIT)

Copyright (c) 2015 Patrick Burtchaell

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.

1.105 sarama-cluster.v2 v2.1.15

1.106 scala-compiler 2.11.0

1.107 schedule-python 0.5.0

1.107.1 Available under license :

The MIT License (MIT)

Copyright (c) 2013 Daniel Bader (http://dbader.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.

The MIT License (MIT)

Copyright (c) 2013 Daniel Bader (http://dbader.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.

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.

1.108 Search Guard SSL 5.6.2-23

1.108.1 Available under license :

floragunn GmbH
Copyright 2015-2016 floragunn GmbH

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

See THIRD-PARTY.txt for additional third party licenses used by this product.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.109 shopify/sarama v1.11.0

1.109.1 Available under license :

Copyright (c) 2013 Evan Huus

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.

1.110 sirupsen/logrus 0.10.0

1.110.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Simon Eskildsen

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.

1.111 spring-boot-starter-tomcat 2.0.3

1.112 tinylib/msgp 1.0.2

1.112.1 Available under license :

Copyright (c) 2014 Philip Hofer
Portions Copyright (c) 2009 The Go Authors (license at http://golang.org) where indicated

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.

1.113 whatwg-fetch 2.0.4

1.113.1 Available under license :

Copyright (c) 2014-2016 GitHub, Inc.

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.

1.114 zlib 1.2.7

1.114.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 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.

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 Lesser 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

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

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.

/* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.7, May 2nd, 2012

Copyright (C) 1995-2012 Jean-loup Gailly and Mark Adler

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.

  Jean-loup Gailly        Mark Adler
  jloup@gzip.org          madler@alumni.caltech.edu


  The data format used by the zlib library is described by RFCs (Request for
  Comments) 1950 to 1952 in the files http://tools.ietf.org/html/rfc1950
  (zlib format), rfc1951 (deflate format) and rfc1952 (gzip format).
*/
----------------------------------------------------------------------------
--                      ZLib for Ada thick binding.                         --
--                                                                          --
--              Copyright (C) 2002-2004 Dmitriy Anisimkov                   --
--                                                                          --
--  This library 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 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       --
--  General Public License for more details.                                --
--                                                                          --
--  You should have received a copy of the GNU General Public License       --
--  along with this library; if not, write to the Free Software Foundation, --
--  Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.          --
--                                                                          --
--  As a special exception, if other files instantiate generics from this   --
--  unit, or you link this unit with other files to produce an executable,  --
--  this  unit  does not  by itself cause  the resulting executable to be   --
--  covered by the GNU General Public License. This exception does not      --
--  however invalidate any other reasons why the executable file  might be  --
--  covered by the  GNU Public License.                                     --
----------------------------------------------------------------------------
```

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries.To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

©2019 Cisco Systems, Inc. All rights reserved.