Open Source Used In C1200 and 1300 Series Switches 4.0.x

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: 78EE117C99-1193302066

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

In your requests please include the following reference number 78EE117C99-1193302066

Contents

1.1 openssl 1.1.1q

1.1.1 Available under license :

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

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

OpenSSL License

```
/* ====================================================================
* Copyright (c) 1998-2019 The OpenSSL Project.  All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. All advertising materials mentioning features or use of this
*    software must display the following acknowledgment:
*    "This product includes software developed by the OpenSSL Project
*    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
```

*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*   endorse or promote products derived from this software without
*   prior written permission. For written permission, please contact
*   openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*   nor may "OpenSSL" appear in their names without prior written
*   permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,

```
* lhash, DES, etc., code; not just the SSL code.  The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*    must display the following acknowledgement:
*    "This product includes cryptographic software written by
*     Eric Young (eay@cryptsoft.com)"
*    The word 'cryptographic' can be left out if the rouines from the library
*    being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*    the apps directory (application code) you must include an acknowledgement:
*    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed.  i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

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

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

You are not responsible for enforcing compliance by third parties to this License.

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

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

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

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

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

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

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

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

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

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

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

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

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

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

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

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

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

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the

library. If this is what you want to do, use the GNU Library General Public License instead of this License.

The "Artistic License"

Preamble

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

Definitions:

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

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

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

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

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

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

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

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

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

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

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

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

d) make other distribution arrangements with the Copyright Holder.

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

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

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

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

d) make other distribution arrangements with the Copyright Holder.

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

form of aggregation, provided that the complete Standard Version of the
interpreter is so embedded.

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

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

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

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

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

The End

1.2 expat 2.4.9

1.2.1 Available under license :

Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper
Copyright (c) 2001-2022 Expat maintainers

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.

1.3 attr 2.5.1-1

1.3.1 Available under license :

Most components of the "attr" package are licensed under
Version 2.1 of the GNU Lesser General Public License (see below).
below.

Some components (as annotated in the source) are licensed
under Version 2 of the GNU General Public License (see COPYING).

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

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

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

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser

General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

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

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

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and

distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

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

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

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

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

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

3. You may opt to apply the terms of the ordinary GNU General Public

License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

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

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials

specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

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

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

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

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

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

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

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

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

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

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

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

NO WARRANTY

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

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE

LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

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

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

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

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

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

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!
Most components of the "attr" package are licensed under
Version 2.1 of the GNU Lesser General Public License (see COPYING.LGPL).

Some components (as annotated in the source) are licensed
under Version 2 of the GNU General Public License (see below),

----

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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.

1.4 busybox 1.34.1

1.4.1 Available under 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
--- 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.

1.5 libcgroup 0.41

1.5.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 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.

[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 St, Fifth Floor, Boston, MA 02110-1301 USA

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James
Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.6 json-c 0.15

1.6.1 Available under license :

Copyright (c) 2009-2012 Eric Haszlakiewicz

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

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

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

Copyright (c) 2004, 2005 Metaparadigm Pte Ltd

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

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

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

1.7 bluez 5.64-1

1.7.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 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.

[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 St, Fifth Floor, Boston, MA 02110-1301 USA

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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.

1.8 usign 2020-05-23

1.8.1 Available under license :

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

/* Edwards curve operations
* Daniel Beer <dlbeer@gmail.com>, 9 Jan 2014
*
* This file is in the public domain.
*/

Found in path(s):
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/ed25519.c
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/ed25519.h
No license file was found, but licenses were detected in source scan.

/* Edwards curve signature system
* Daniel Beer <dlbeer@gmail.com>, 22 Apr 2014
*
* This file is in the public domain.
*/

Found in path(s):
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/edsign.h
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/edsign.c
No license file was found, but licenses were detected in source scan.

/* Arithmetic in prime fields
* Daniel Beer <dlbeer@gmail.com>, 10 Jan 2014
*
* This file is in the public domain.
*/

Found in path(s):
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/fprime.c
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/fprime.h
No license file was found, but licenses were detected in source scan.

Package: usign
Version: 2020-05-23-f1f65026-1
Depends: libc, librt, libpthread, libubox20220515
Source: package/system/usign
SourceName: usign
License: ISC

Section: base
SourceDateEpoch: 1590233892
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 13584
Description: OpenWrt signature verification utility

Found in path(s):
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/ipkg-aarch64_cortex-a72/usign/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* usign - tiny signify replacement
*
* Copyright (C) 2015 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/main.c
No license file was found, but licenses were detected in source scan.

/* Arithmetic mod p = 2^255-19
* Daniel Beer <dlbeer@gmail.com>, 5 Jan 2014
*
* This file is in the public domain.
*/

Found in path(s):
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/f25519.c
No license file was found, but licenses were detected in source scan.

/* Arithmetic mod p = 2^255-19
* Daniel Beer <dlbeer@gmail.com>, 8 Jan 2014

*
* This file is in the public domain.
*/

Found in path(s):
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/f25519.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Felix Fietkau <nbd@openwrt.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.
*/
/* SHA512
* Daniel Beer <dlbeer@gmail.com>, 22 Apr 2014
*
* This file is in the public domain.
*/

Found in path(s):
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/sha512.c
* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/sha512.h
No license file was found, but licenses were detected in source scan.

/*
* base64 - libubox base64 functions
*
* Copyright (C) 2015 Felix Fietkau <nbd@openwrt.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.
*/
/*
* Copyright (c) 1996 by Internet Software Consortium.
*
* 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 INTERNET SOFTWARE CONSORTIUM DISCLAIMS
* ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE
* CONSORTIUM 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.
*/
/*
* Portions Copyright (c) 1995 by International Business Machines, Inc.
*
* International Business Machines, Inc. (hereinafter called IBM) grants
* permission under its copyrights to use, copy, modify, and distribute this
* Software with or without fee, provided that the above copyright notice and
* all paragraphs of this notice appear in all copies, and that the name of IBM
* not be used in connection with the marketing of any product incorporating
* the Software or modifications thereof, without specific, written prior
* permission.
*
* To the extent it has a right to do so, IBM grants an immunity from suit
* under its patents, if any, for the use, sale or manufacture of products to
* the extent that such products are used for performing Domain Name System
* dynamic updates in TCP/IP networks by means of the Software. No immunity is
* granted for any product per se or for any other function of any product.
*
* THE SOFTWARE IS PROVIDED "AS IS", AND IBM DISCLAIMS ALL WARRANTIES,
* INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
* PARTICULAR PURPOSE. IN NO EVENT SHALL IBM BE LIABLE FOR ANY SPECIAL,
* DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER ARISING
* OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE, EVEN
* IF IBM IS APPRISED OF THE POSSIBILITY OF SUCH DAMAGES.
*/

Found in path(s):

* /opt/cola/permits/1601456330_1679570368.625229/0/usign-2020-05-23-f1f65026-zip/usign-2020-05-23-f1f65026/base64.c

1.9 mbed-tls 2.28.1

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.

1.10 libical 3.0.9

1.10.1 Available under license :

This package was debianized by Fathi Boudra <fboudra@free.fr> on
Wed, 27 Dec 2006 12:19:06 +0100.

It was downloaded from http://www.aurore.net/projects/libical/

This is a modified version of the original libical project from softwarestudio
and the original author is Eric Busboom <eric@civicknowledge.com>.

Upstream Authors:
John Gray <gray@agora-net.com>
Andrea Campi <a.campi@inet.it>
Omar Kilani <omar@tinysofa.org>

Copyright for generated icalderivedproperties:
(C) 1999 Eric Busboom <eric@civicknowledge.com>

Copyright for generated restrictionrecords and parameterrestrictions:
(C) 1999 Graham Davison <g.m.davison@computer.org>

Copyright for sspm based on code from the mimelite distribution:
(C) 1994 Gisle Hannemyr <gisle@oslonett.no>
(C) 2000 Eric Busboom <eric@civicknowledge.com>

Copyright for snprintf:
(C) 1997 Theo de Raadt

Copyright for astime and caldate:
(C) 1986-2000 Hiram Clawson

Copyright for icalattach:
(C) 2002 Andrea Campi <a.campi@inet.it>

Copyright for icaltimezone:
(C) 2001 Damon Chaplin

Copyright for icalarray:
(C) 2001 Ximian Inc.

astime and caldate 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.

Neither name of The Museum of Hiram 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.

snprintf license:

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.

sspm license:

Permission is granted to hack, make and distribute copies of this program as long as this copyright notice is not removed.

libical license:

The code and datafiles in this distribution are licensed under the Mozilla Public License. See http://www.mozilla.org/NPL/MPL-1.0.html for a copy of the license. Alternately, you may use libical under the terms of the GNU Library General Public License. See http://www.fsf.org/copyleft/lesser.html for a copy of the LGPL.

This dual license ensures that the library can be incorporated into both proprietary code and GPL'd programs, and will benefit from improvements made by programmers in both realms. I will only accept changes into my version of the library if they are similarly dual-licensed.

This program is free software; you can redistribute it and/or modify it under the terms of either:

The LGPL as published by the Free Software Foundation, version 2.1, available at: http://www.fsf.org/copyleft/lesser.html

Or:

The Mozilla Public License Version 1.0. You may obtain a copy of the License at http://www.mozilla.org/MPL/

MPL 1.0 license:

MOZILLA PUBLIC LICENSE
Version 1.0

----

1. Definitions.

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.

1.9. ``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.

1.11. ``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 a list of 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'' 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 fifty percent (50%) or more 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) to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Initial Developer, to make, have made, use and sell (``Utilize'') the Original Code (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Original Code (or portions thereof) and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

2.2. Contributor Grant.
Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) 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 or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Contributor, to Utilize the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to Utilize the Contributor Version (or portions thereof), and not to any greater extent that may be necessary to Utilize further Modifications or combinations.

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 You have knowledge that a party claims an intellectual property right in particular functionality or code (or its utilization under this License), you 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 you obtain such knowledge after You make Your Modification available as described in Section 3.2, You shall promptly modify the LEGAL file in all copies You make 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 Your Modification is an application programming interface and You own or control patents which are reasonably necessary to implement that API, you must also include this information in the LEGAL file.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code, and this License in any documentation for the Source Code, where You describe recipients' rights relating to Covered Code. If You created one or more Modification(s), You may add your name as a Contributor to the notice described in Exhibit A. 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 file) where a user would be likely to look for such a notice. 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 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 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'', ``NPL'' or
any confusingly similar phrase do not appear anywhere in your 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.

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.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT
(INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL THE INITIAL
DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE,
OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO YOU OR ANY OTHER
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 THAT 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: (a) unless otherwise agreed in writing, all disputes relating to this License (excepting any dispute relating to intellectual property rights) shall be subject to final and binding arbitration, with the losing party paying all costs of arbitration; (b) any arbitration relating to this Agreement shall be held in Santa Clara County, California, under the auspices of JAMS/EndDispute; and (c) any litigation relating to this Agreement 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.

Except in cases where another Contributor has failed to comply with
Section 3.4, You are responsible for damages arising, directly or
indirectly, out of Your utilization of rights under this License, based
on the number of copies of Covered Code you made available, the
revenues you received from utilizing such rights, and other relevant
factors. You agree to work with affected parties to distribute
responsibility on an equitable basis.

EXHIBIT A.

``The contents of this file are subject to the Mozilla Public License
Version 1.0 (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.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): ______________________________________."

LGPL license:

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 Lesser General Public
License along with this package; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

On Debian systems, the complete text of the GNU Lesser General
Public License can be found in `/usr/share/common-licenses/LGPL'.

The Debian packaging is (C) 2006, Fathi Boudra <fboudra@free.fr> and

is licensed under the GPL, see `/usr/share/common-licenses/GPL'.
libical is distributed under two licenses.
You may choose the terms of either:

* The Mozilla Public License (MPL) v2.0

or

* The GNU Lesser General Public License (LGPL) v2.1

Software distributed under these licenses 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 libical.

The Initial Developer of the Original Code is Eric Busboom

All Rights Reserved.

Contributor(s): See individual source files.
libical is distributed under two licenses.
You may choose the terms of either:

* The Mozilla Public License (MPL) v2.0

or

* The GNU Lesser General Public License (LGPL) v2.1

Software distributed under these licenses 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.
Libical is distributed under both the LGPL and the MPL. The MPL
notice, reproduced below, covers the use of either of the licenses.

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 https://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.

Libical is Copyright (C) 2010-2017 Klaralvdalens Datakonsult AB.

You may use, distribute and copy Libical under the terms of the
GNU Lesser General Public License version 2.1, which is displayed below.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 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.

[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 St, Fifth Floor, Boston, MA 02110-1301 USA

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James
Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.11 fwtool 2019-11-12

1.11.1 Available under license :

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

/*
* Copyright (C) 2016 Felix Fietkau <nbd@nbd.name>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License version 2
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601412993_1679570273.2046802/0/fwtool-2019-11-12-8f7fe925-zip/fwtool-2019-11-12-8f7fe925/fwimage.h
* /opt/cola/permits/1601412993_1679570273.2046802/0/fwtool-2019-11-12-8f7fe925-zip/fwtool-2019-11-12-8f7fe925/fwtool.c
No license file was found, but licenses were detected in source scan.

/*

* utils - misc libubox utility functions
*
* Copyright (C) 2012 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601412993_1679570273.2046802/0/fwtool-2019-11-12-8f7fe925-zip/fwtool-2019-11-12-8f7fe925/utils.h
No license file was found, but licenses were detected in source scan.

Package: fwtool
Version: 2019-11-12-8f7fe925-1
Depends: libc, librt, libpthread
Source: package/system/fwtool
SourceName: fwtool
License: GPL-2.0
Section: utils
SourceDateEpoch: 1573768411
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 5354
Description: Utility for appending and extracting firmware metadata and signatures

Found in path(s):
* /opt/cola/permits/1601412993_1679570273.2046802/0/fwtool-2019-11-12-8f7fe925-zip/fwtool-2019-11-12-8f7fe925/ipkg-aarch64_cortex-a72/fwtool/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 Felix Fietkau <nbd@nbd.name>
*
* Based on busybox code:
*   CRC32 table fill function
*   Copyright (C) 2006 by Rob Sullivan <cogito.ergo.cogito@gmail.com>
*
* This program is free software; you can redistribute it and/or modify

* it under the terms of the GNU General Public License version 2
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601412993_1679570273.2046802/0/fwtool-2019-11-12-8f7fe925-zip/fwtool-2019-11-12-8f7fe925/crc32.h

1.12 net-snmp 2.9.1

1.12.1 Available under license :

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

```
IANAifType-MIB DEFINITIONS ::= BEGIN

  IMPORTS
      MODULE-IDENTITY, mib-2      FROM SNMPv2-SMI
      TEXTUAL-CONVENTION          FROM SNMPv2-TC;

  ianaifType MODULE-IDENTITY
      LAST-UPDATED "201205170000Z"  -- May 17, 2012
      ORGANIZATION "IANA"
      CONTACT-INFO "       Internet Assigned Numbers Authority

                   Postal: ICANN
                           4676 Admiralty Way, Suite 330
                           Marina del Rey, CA 90292

                   Tel:    +1 310 823 9358
                   E-Mail: iana&amp;iana.org"

      DESCRIPTION  "This MIB module defines the IANAifType Textual
                   Convention, and thus the enumerated values of
                   the ifType object defined in MIB-II's ifTable."

      REVISION     "201205170000Z"  -- May 17, 2012
      DESCRIPTION  "Registration of new IANAifType 272."

      REVISION     "201201110000Z"  -- January 11, 2012
      DESCRIPTION  "Registration of new IANAifTypes 266-271."

      REVISION     "201112180000Z"  -- December 18, 2011
```

DESCRIPTION "Registration of new IANAifTypes 263-265."

REVISION   "201110260000Z" -- October 26, 2011
DESCRIPTION "Registration of new IANAifType 262."

REVISION   "201109070000Z" -- September 7, 2011
DESCRIPTION "Registration of new IANAifTypes 260 and 261."

REVISION   "201107220000Z" -- July 22, 2011
DESCRIPTION "Registration of new IANAifType 259."

REVISION   "201106030000Z" -- June 03, 2011
DESCRIPTION "Registration of new IANAifType 258."

REVISION   "201009210000Z" -- September 21, 2010
DESCRIPTION "Registration of new IANAifTypes 256 and 257."

REVISION   "201007210000Z" -- July 21, 2010
DESCRIPTION "Registration of new IANAifType 255."

REVISION   "201002110000Z" -- February 11, 2010
DESCRIPTION "Registration of new IANAifType 254."

REVISION   "201002080000Z" -- February 08, 2010
DESCRIPTION "Registration of new IANAifTypes 252 and 253."

REVISION   "200905060000Z" -- May 06, 2009
DESCRIPTION "Registration of new IANAifType 251."

REVISION   "200902060000Z" -- February 06, 2009
DESCRIPTION "Registration of new IANAtunnelType 15."

REVISION   "200810090000Z" -- October 09, 2008
DESCRIPTION "Registration of new IANAifType 250."

REVISION   "200808120000Z" -- August 12, 2008
DESCRIPTION "Registration of new IANAifType 249."

REVISION   "200807220000Z" -- July 22, 2008
DESCRIPTION "Registration of new IANAifTypes 247 and 248."

REVISION   "200806240000Z" -- June 24, 2008
DESCRIPTION "Registration of new IANAifType 246."

REVISION   "200805290000Z" -- May 29, 2008
DESCRIPTION "Registration of new IANAifType 245."

REVISION   "200709130000Z" -- September 13, 2007

```
DESCRIPTION  "Registration of new IANAifTypes 243 and 244."

REVISION     "200705290000Z"  -- May 29, 2007
DESCRIPTION  "Changed the description for IANAifType 228."

REVISION     "200703080000Z"  -- March 08, 2007
DESCRIPTION  "Registration of new IANAifType 242."

REVISION     "200701230000Z"  -- January 23, 2007
DESCRIPTION  "Registration of new IANAifTypes 239, 240, and 241."

REVISION     "200610170000Z"  -- October 17, 2006
DESCRIPTION  "Deprecated/Obsoleted IANAifType 230.  Registration of
             IANAifType 238."

REVISION     "200609250000Z"  -- September 25, 2006
DESCRIPTION  "Changed the description for IANA ifType
             184 and added new IANA ifType 237."

REVISION     "200608170000Z"  -- August 17, 2006
DESCRIPTION  "Changed the descriptions for IANAifTypes
             20 and 21."

REVISION     "200608110000Z"  -- August 11, 2006
DESCRIPTION  "Changed the descriptions for IANAifTypes
             7, 11, 62, 69, and 117."

REVISION     "200607250000Z"  -- July 25, 2006
DESCRIPTION  "Registration of new IANA ifType 236."

REVISION     "200606140000Z"  -- June 14, 2006
DESCRIPTION  "Registration of new IANA ifType 235."

REVISION     "200603310000Z"  -- March 31, 2006
DESCRIPTION  "Registration of new IANA ifType 234."

REVISION     "200603300000Z"  -- March 30, 2006
DESCRIPTION  "Registration of new IANA ifType 233."

REVISION     "200512220000Z"  -- December 22, 2005
DESCRIPTION  "Registration of new IANA ifTypes 231 and 232."

REVISION     "200510100000Z"  -- October 10, 2005
DESCRIPTION  "Registration of new IANA ifType 230."

REVISION     "200509090000Z"  -- September 09, 2005
DESCRIPTION  "Registration of new IANA ifType 229."
```

REVISION   "200505270000Z" -- May 27, 2005
DESCRIPTION "Registration of new IANA ifType 228."

REVISION   "200503030000Z" -- March 3, 2005
DESCRIPTION "Added the IANAtunnelType TC and deprecated
IANAifType sixToFour (215) per RFC4087."

REVISION   "200411220000Z" -- November 22, 2004
DESCRIPTION "Registration of new IANA ifType 227 per RFC4631."

REVISION   "200406170000Z" -- June 17, 2004
DESCRIPTION "Registration of new IANA ifType 226."

REVISION   "200405120000Z" -- May 12, 2004
DESCRIPTION "Added description for IANAifType 6, and
changed the descriptions for IANAifTypes
180, 181, and 182."

REVISION   "200405070000Z" -- May 7, 2004
DESCRIPTION "Registration of new IANAifType 225."

REVISION   "200308250000Z" -- Aug 25, 2003
DESCRIPTION "Deprecated IANAifTypes 7 and 11. Obsoleted
IANAifTypes 62, 69, and 117. ethernetCsmacd (6)
should be used instead of these values"

REVISION   "200308180000Z" -- Aug 18, 2003
DESCRIPTION "Registration of new IANAifType
224."

REVISION   "200308070000Z" -- Aug 7, 2003
DESCRIPTION "Registration of new IANAifTypes
222 and 223."

REVISION   "200303180000Z" -- Mar 18, 2003
DESCRIPTION "Registration of new IANAifType
221."

REVISION   "200301130000Z" -- Jan 13, 2003
DESCRIPTION "Registration of new IANAifType
220."

REVISION   "200210170000Z" -- Oct 17, 2002
DESCRIPTION "Registration of new IANAifType
219."

REVISION   "200207160000Z" -- Jul 16, 2002
DESCRIPTION "Registration of new IANAifTypes

```
        217 and 218."

REVISION     "200207100000Z"  -- Jul 10, 2002
DESCRIPTION  "Registration of new IANAifTypes
        215 and 216."

REVISION     "200206190000Z"  -- Jun 19, 2002
DESCRIPTION  "Registration of new IANAifType
        214."

REVISION     "200201040000Z"  -- Jan 4, 2002
DESCRIPTION  "Registration of new IANAifTypes
        211, 212 and 213."

REVISION     "200112200000Z"  -- Dec 20, 2001
DESCRIPTION  "Registration of new IANAifTypes
        209 and 210."

REVISION     "200111150000Z"  -- Nov 15, 2001
DESCRIPTION  "Registration of new IANAifTypes
        207 and 208."


REVISION     "200111060000Z"  -- Nov 6, 2001
DESCRIPTION  "Registration of new IANAifType
        206."


REVISION     "200111020000Z"  -- Nov 2, 2001
DESCRIPTION  "Registration of new IANAifType
        205."


REVISION     "200110160000Z"  -- Oct 16, 2001
DESCRIPTION  "Registration of new IANAifTypes
        199, 200, 201, 202, 203, and 204."


REVISION     "200109190000Z"  -- Sept 19, 2001
DESCRIPTION  "Registration of new IANAifType
        198."

REVISION     "200105110000Z"  -- May 11, 2001
DESCRIPTION  "Registration of new IANAifType
        197."


REVISION     "200101120000Z"  -- Jan 12, 2001
```

```
DESCRIPTION  "Registration of new IANAifTypes
             195 and 196."

REVISION     "200012190000Z"  -- Dec 19, 2000
DESCRIPTION  "Registration of new IANAifTypes
             193 and 194."

REVISION     "200012070000Z"  -- Dec 07, 2000
DESCRIPTION  "Registration of new IANAifTypes
             191 and 192."

REVISION     "200012040000Z"  -- Dec 04, 2000
DESCRIPTION  "Registration of new IANAifType
             190."

REVISION     "200010170000Z"  -- Oct 17, 2000
DESCRIPTION  "Registration of new IANAifTypes
             188 and 189."

REVISION     "200010020000Z"  -- Oct 02, 2000
DESCRIPTION  "Registration of new IANAifType 187."

REVISION     "200009010000Z"  -- Sept 01, 2000
DESCRIPTION  "Registration of new IANAifTypes
             184, 185, and 186."

REVISION     "200008240000Z"  -- Aug 24, 2000
DESCRIPTION  "Registration of new IANAifType 183."

REVISION     "200008230000Z"  -- Aug 23, 2000
DESCRIPTION  "Registration of new IANAifTypes
             174-182."

REVISION     "200008220000Z"  -- Aug 22, 2000
DESCRIPTION  "Registration of new IANAifTypes 170,
             171, 172 and 173."

REVISION     "200004250000Z"  -- Apr 25, 2000
DESCRIPTION  "Registration of new IANAifTypes 168 and 169."

REVISION     "200003060000Z"  -- Mar 6, 2000
DESCRIPTION  "Fixed a missing semi-colon in the IMPORT.
             Also cleaned up the REVISION log a bit.
             It is not complete, but from now on it will
             be maintained and kept up to date with each
             change to this MIB module."

REVISION     "199910081430Z"  -- Oct 08, 1999
```

```
   DESCRIPTION  "Include new name assignments up to cnr(85).
              This is the first version available via the WWW
              at: ftp://ftp.isi.edu/mib/ianaiftype.mib"

   REVISION     "199401310000Z"  -- Jan 31, 1994
   DESCRIPTION  "Initial version of this MIB as published in
              RFC 1573."

   ::= { mib-2 30 }


IANAifType ::= TEXTUAL-CONVENTION
   STATUS       current
   DESCRIPTION
        "This data type is used as the syntax of the ifType
        object in the (updated) definition of MIB-II's
        ifTable.

        The definition of this textual convention with the
        addition of newly assigned values is published
        periodically by the IANA, in either the Assigned
        Numbers RFC, or some derivative of it specific to
        Internet Network Management number assignments.  (The
        latest arrangements can be obtained by contacting the
        IANA.)

        Requests for new values should be made to IANA via
        email (iana&amp;iana.org).

        The relationship between the assignment of ifType
        values and of OIDs to particular media-specific MIBs
        is solely the purview of IANA and is subject to change
        without notice.  Quite often, a media-specific MIB's
        OID-subtree assignment within MIB-II's 'transmission'
        subtree will be the same as its ifType value.
        However, in some circumstances this will not be the
        case, and implementors must not pre-assume any
        specific relationship between ifType values and
        transmission subtree OIDs."
   SYNTAX  INTEGER {
          other(1),          -- none of the following
          regular1822(2),
          hdh1822(3),
          ddnX25(4),
          rfc877x25(5),
          ethernetCsmacd(6), -- for all ethernet-like interfaces,
                             -- regardless of speed, as per RFC3635
          iso88023Csmacd(7), -- Deprecated via RFC3635
```

```
                  -- ethernetCsmacd (6) should be used instead
iso88024TokenBus(8),
iso88025TokenRing(9),
iso88026Man(10),
starLan(11), -- Deprecated via RFC3635
            -- ethernetCsmacd (6) should be used instead
proteon10Mbit(12),
proteon80Mbit(13),
hyperchannel(14),
fddi(15),
lapb(16),
sdlc(17),
ds1(18),           -- DS1-MIB
e1(19),            -- Obsolete see DS1-MIB
basicISDN(20),             -- no longer used
                           -- see also RFC2127
primaryISDN(21),           -- no longer used
                           -- see also RFC2127
propPointToPointSerial(22), -- proprietary serial
ppp(23),
softwareLoopback(24),
eon(25),           -- CLNP over IP
ethernet3Mbit(26),
nsip(27),          -- XNS over IP
slip(28),          -- generic SLIP
ultra(29),         -- ULTRA technologies
ds3(30),           -- DS3-MIB
sip(31),           -- SMDS, coffee
frameRelay(32),    -- DTE only.
rs232(33),
para(34),          -- parallel-port
arcnet(35),        -- arcnet
arcnetPlus(36),    -- arcnet plus
atm(37),           -- ATM cells
miox25(38),
sonet(39),         -- SONET or SDH
x25ple(40),
iso88022llc(41),
localTalk(42),
smdsDxi(43),
frameRelayService(44),  -- FRNETSERV-MIB
v35(45),
hssi(46),
hippi(47),
modem(48),         -- Generic modem
aal5(49),          -- AAL5 over ATM
sonetPath(50),
sonetVT(51),
```

```
smdsIcip(52),       -- SMDS InterCarrier Interface
propVirtual(53),    -- proprietary virtual/internal
propMultiplexor(54),-- proprietary multiplexing
ieee80212(55),      -- 100BaseVG
fibreChannel(56),   -- Fibre Channel
hippiInterface(57), -- HIPPI interfaces
frameRelayInterconnect(58), -- Obsolete, use either
                    -- frameRelay(32) or
                    -- frameRelayService(44).
aflane8023(59),     -- ATM Emulated LAN for 802.3
aflane8025(60),     -- ATM Emulated LAN for 802.5
cctEmul(61),        -- ATM Emulated circuit
fastEther(62),      -- Obsoleted via RFC3635
                    -- ethernetCsmacd (6) should be used instead
isdn(63),           -- ISDN and X.25
v11(64),            -- CCITT V.11/X.21
v36(65),            -- CCITT V.36
g703at64k(66),      -- CCITT G703 at 64Kbps
g703at2mb(67),      -- Obsolete see DS1-MIB
qllc(68),           -- SNA QLLC
fastEtherFX(69),    -- Obsoleted via RFC3635
                    -- ethernetCsmacd (6) should be used instead
channel(70),        -- channel
ieee80211(71),      -- radio spread spectrum
ibm370parChan(72),  -- IBM System 360/370 OEMI Channel
escon(73),          -- IBM Enterprise Systems Connection
dlsw(74),           -- Data Link Switching
isdns(75),          -- ISDN S/T interface
isdnu(76),          -- ISDN U interface
lapd(77),           -- Link Access Protocol D
ipSwitch(78),       -- IP Switching Objects
rsrb(79),           -- Remote Source Route Bridging
atmLogical(80),     -- ATM Logical Port
ds0(81),            -- Digital Signal Level 0
ds0Bundle(82),      -- group of ds0s on the same ds1
bsc(83),            -- Bisynchronous Protocol
async(84),          -- Asynchronous Protocol
cnr(85),            -- Combat Net Radio
iso88025Dtr(86),    -- ISO 802.5r DTR
eplrs(87),          -- Ext Pos Loc Report Sys
arap(88),           -- Appletalk Remote Access Protocol
propCnls(89),       -- Proprietary Connectionless Protocol
hostPad(90),        -- CCITT-ITU X.29 PAD Protocol
termPad(91),        -- CCITT-ITU X.3 PAD Facility
frameRelayMPI(92),  -- Multiproto Interconnect over FR
x213(93),           -- CCITT-ITU X213
adsl(94),           -- Asymmetric Digital Subscriber Loop
radsl(95),          -- Rate-Adapt. Digital Subscriber Loop
```

```
sdsl(96),          -- Symmetric Digital Subscriber Loop
vdsl(97),          -- Very H-Speed Digital Subscrib. Loop
iso88025CRFPInt(98), -- ISO 802.5 CRFP
myrinet(99),       -- Myricom Myrinet
voiceEM(100),      -- voice recEive and transMit
voiceFXO(101),     -- voice Foreign Exchange Office
voiceFXS(102),     -- voice Foreign Exchange Station
voiceEncap(103),   -- voice encapsulation
voiceOverIp(104),  -- voice over IP encapsulation
atmDxi(105),       -- ATM DXI
atmFuni(106),      -- ATM FUNI
atmIma (107),      -- ATM IMA
pppMultilinkBundle(108), -- PPP Multilink Bundle
ipOverCdlc (109),  -- IBM ipOverCdlc
ipOverClaw (110),  -- IBM Common Link Access to Workstn
stackToStack (111), -- IBM stackToStack
virtualIpAddress (112), -- IBM VIPA
mpc (113),         -- IBM multi-protocol channel support
ipOverAtm (114),   -- IBM ipOverAtm
iso88025Fiber (115), -- ISO 802.5j Fiber Token Ring
tdlc (116),        -- IBM twinaxial data link control
gigabitEthernet (117), -- Obsoleted via RFC3635
                   -- ethernetCsmacd (6) should be used instead
hdlc (118),        -- HDLC
lapf (119),        -- LAP F
v37 (120),         -- V.37
x25mlp (121),      -- Multi-Link Protocol
x25huntGroup (122), -- X25 Hunt Group
transpHdlc (123),  -- Transp HDLC
interleave (124),  -- Interleave channel
fast (125),        -- Fast channel
ip (126),          -- IP (for APPN HPR in IP networks)
docsCableMaclayer (127),  -- CATV Mac Layer
docsCableDownstream (128), -- CATV Downstream interface
docsCableUpstream (129),  -- CATV Upstream interface
a12MppSwitch (130), -- Avalon Parallel Processor
tunnel (131),      -- Encapsulation interface
coffee (132),      -- coffee pot
ces (133),         -- Circuit Emulation Service
atmSubInterface (134), -- ATM Sub Interface
l2vlan (135),      -- Layer 2 Virtual LAN using 802.1Q
l3ipvlan (136),    -- Layer 3 Virtual LAN using IP
l3ipxvlan (137),   -- Layer 3 Virtual LAN using IPX
digitalPowerline (138), -- IP over Power Lines
mediaMailOverIp (139), -- Multimedia Mail over IP
dtm (140),         -- Dynamic syncronous Transfer Mode
dcn (141),    -- Data Communications Network
ipForward (142),   -- IP Forwarding Interface
```

msdsl (143), -- Multi-rate Symmetric DSL
ieee1394 (144), -- IEEE1394 High Performance Serial Bus
if-gsn (145), -- HIPPI-6400
dvbRccMacLayer (146), -- DVB-RCC MAC Layer
dvbRccDownstream (147), -- DVB-RCC Downstream Channel
dvbRccUpstream (148), -- DVB-RCC Upstream Channel
atmVirtual (149), -- ATM Virtual Interface
mplsTunnel (150), -- MPLS Tunnel Virtual Interface
srp (151), -- Spatial Reuse Protocol
voiceOverAtm (152), -- Voice Over ATM
voiceOverFrameRelay (153), -- Voice Over Frame Relay
idsl (154), -- Digital Subscriber Loop over ISDN
compositeLink (155), -- Avici Composite Link Interface
ss7SigLink (156), -- SS7 Signaling Link
propWirelessP2P (157), -- Prop. P2P wireless interface
frForward (158), -- Frame Forward Interface
rfc1483 (159), -- Multiprotocol over ATM AAL5
usb (160), -- USB Interface
ieee8023adLag (161), -- IEEE 802.3ad Link Aggregate
bgppolicyaccounting (162), -- BGP Policy Accounting
frf16MfrBundle (163), -- FRF .16 Multilink Frame Relay
h323Gatekeeper (164), -- H323 Gatekeeper
h323Proxy (165), -- H323 Voice and Video Proxy
mpls (166), -- MPLS
mfSigLink (167), -- Multi-frequency signaling link
hdsl2 (168), -- High Bit-Rate DSL - 2nd generation
shdsl (169), -- Multirate HDSL2
ds1FDL (170), -- Facility Data Link 4Kbps on a DS1
pos (171), -- Packet over SONET/SDH Interface
dvbAsiIn (172), -- DVB-ASI Input
dvbAsiOut (173), -- DVB-ASI Output
plc (174), -- Power Line Communtications
nfas (175), -- Non Facility Associated Signaling
tr008 (176), -- TR008
gr303RDT (177), -- Remote Digital Terminal
gr303IDT (178), -- Integrated Digital Terminal
isup (179), -- ISUP
propDocsWirelessMaclayer (180), -- Cisco proprietary Maclayer
propDocsWirelessDownstream (181), -- Cisco proprietary Downstream
propDocsWirelessUpstream (182), -- Cisco proprietary Upstream
hiperlan2 (183), -- HIPERLAN Type 2 Radio Interface
propBWAp2Mp (184), -- PropBroadbandWirelessAccesspt2multipt
    -- use of this iftype for IEEE 802.16 WMAN
    -- interfaces as per IEEE Std 802.16f is
    -- deprecated and ifType 237 should be used instead.
sonetOverheadChannel (185), -- SONET Overhead Channel
digitalWrapperOverheadChannel (186), -- Digital Wrapper
aal2 (187), -- ATM adaptation layer 2

radioMAC (188), -- MAC layer over radio links
atmRadio (189), -- ATM over radio links
imt (190), -- Inter Machine Trunks
mvl (191), -- Multiple Virtual Lines DSL
reachDSL (192), -- Long Reach DSL
frDlciEndPt (193), -- Frame Relay DLCI End Point
atmVciEndPt (194), -- ATM VCI End Point
opticalChannel (195), -- Optical Channel
opticalTransport (196), -- Optical Transport
propAtm (197), -- Proprietary ATM
voiceOverCable (198), -- Voice Over Cable Interface
infiniband (199), -- Infiniband
teLink (200), -- TE Link
q2931 (201), -- Q.2931
virtualTg (202), -- Virtual Trunk Group
sipTg (203), -- SIP Trunk Group
sipSig (204), -- SIP Signaling
docsCableUpstreamChannel (205), -- CATV Upstream Channel
econet (206), -- Acorn Econet
pon155 (207), -- FSAN 155Mb Symetrical PON interface
pon622 (208), -- FSAN622Mb Symetrical PON interface
bridge (209), -- Transparent bridge interface
linegroup (210), -- Interface common to multiple lines
voiceEMFGD (211), -- voice E&M Feature Group D
voiceFGDEANA (212), -- voice FGD Exchange Access North American
voiceDID (213), -- voice Direct Inward Dialing
mpegTransport (214), -- MPEG transport interface
sixToFour (215), -- 6to4 interface (DEPRECATED)
gtp (216), -- GTP (GPRS Tunneling Protocol)
pdnEtherLoop1 (217), -- Paradyne EtherLoop 1
pdnEtherLoop2 (218), -- Paradyne EtherLoop 2
opticalChannelGroup (219), -- Optical Channel Group
homepna (220), -- HomePNA ITU-T G.989
gfp (221), -- Generic Framing Procedure (GFP)
ciscoISLvlan (222), -- Layer 2 Virtual LAN using Cisco ISL
actelisMetaLOOP (223), -- Acteleis proprietary MetaLOOP High Speed Link
fcipLink (224), -- FCIP Link
rpr (225), -- Resilient Packet Ring Interface Type
qam (226), -- RF Qam Interface
lmp (227), -- Link Management Protocol
cblVectaStar (228), -- Cambridge Broadband Networks Limited VectaStar
docsCableMCmtsDownstream (229), -- CATV Modular CMTS Downstream Interface
adsl2 (230), -- Asymmetric Digital Subscriber Loop Version 2
    -- (DEPRECATED/OBSOLETED - please use adsl2plus 238 instead)
macSecControlledIF (231), -- MACSecControlled
macSecUncontrolledIF (232), -- MACSecUncontrolled
aviciOpticalEther (233), -- Avici Optical Ethernet Aggregate
atmbond (234), -- atmbond

```
            voiceFGDOS (235), -- voice FGD Operator Services
            mocaVersion1 (236), -- MultiMedia over Coax Alliance (MoCA) Interface
                  -- as documented in information provided privately to IANA
            ieee80216WMAN (237), -- IEEE 802.16 WMAN interface
            adsl2plus (238), -- Asymmetric Digital Subscriber Loop Version 2,
                     -- Version 2 Plus and all variants
            dvbRcsMacLayer (239), -- DVB-RCS MAC Layer
            dvbTdm (240), -- DVB Satellite TDM
            dvbRcsTdma (241), -- DVB-RCS TDMA
            x86Laps (242), -- LAPS based on ITU-T X.86/Y.1323
            wwanPP (243), -- 3GPP WWAN
            wwanPP2 (244), -- 3GPP2 WWAN
            voiceEBS (245), -- voice P-phone EBS physical interface
            ifPwType (246), -- Pseudowire interface type
            ilan (247), -- Internal LAN on a bridge per IEEE 802.1ap
            pip (248), -- Provider Instance Port on a bridge per IEEE 802.1ah PBB
            aluELP (249), -- Alcatel-Lucent Ethernet Link Protection
            gpon (250), -- Gigabit-capable passive optical networks (G-PON) as per ITU-T G.948
            vdsl2 (251), -- Very high speed digital subscriber line Version 2 (as per ITU-T Recommendation
G.993.2)
            capwapDot11Profile (252), -- WLAN Profile Interface
            capwapDot11Bss (253), -- WLAN BSS Interface
            capwapWtpVirtualRadio (254), -- WTP Virtual Radio Interface
            bits (255), -- bitsport
            docsCableUpstreamRfPort (256), -- DOCSIS CATV Upstream RF Port
            cableDownstreamRfPort (257), -- CATV downstream RF port
            vmwareVirtualNic (258), -- VMware Virtual Network Interface
            ieee802154 (259), -- IEEE 802.15.4 WPAN interface
            otnOdu (260), -- OTN Optical Data Unit
            otnOtu (261), -- OTN Optical channel Transport Unit
            ifVfiType (262), -- VPLS Forwarding Instance Interface Type
            g9981 (263), -- G.998.1 bonded interface
            g9982 (264), -- G.998.2 bonded interface
            g9983 (265), -- G.998.3 bonded interface
            aluEpon (266), -- Ethernet Passive Optical Networks (E-PON)
            aluEponOnu (267), -- EPON Optical Network Unit
            aluEponPhysicalUni (268), -- EPON physical User to Network interface
            aluEponLogicalLink (269), -- The emulation of a point-to-point link over the EPON layer
            aluGponOnu (270), -- GPON Optical Network Unit
            aluGponPhysicalUni (271), -- GPON physical User to Network interface
            vmwareNicTeam (272) -- VMware NIC Team
            }

IANAtunnelType ::= TEXTUAL-CONVENTION
  STATUS     current
  DESCRIPTION
       "The encapsulation method used by a tunnel. The value
       direct indicates that a packet is encapsulated
```

directly within a normal IP header, with no
intermediate header, and unicast to the remote tunnel
endpoint (e.g., an RFC 2003 IP-in-IP tunnel, or an RFC
1933 IPv6-in-IPv4 tunnel). The value minimal indicates
that a Minimal Forwarding Header (RFC 2004) is
inserted between the outer header and the payload
packet. The value UDP indicates that the payload
packet is encapsulated within a normal UDP packet
(e.g., RFC 1234).

The values sixToFour, sixOverFour, and isatap
indicates that an IPv6 packet is encapsulated directly
within an IPv4 header, with no intermediate header,
and unicast to the destination determined by the 6to4,
6over4, or ISATAP protocol.

The remaining protocol-specific values indicate that a
header of the protocol of that name is inserted
between the outer header and the payload header.

The assignment policy for IANAtunnelType values is
identical to the policy for assigning IANAifType
values."

```
SYNTAX     INTEGER {
        other(1),        -- none of the following
        direct(2),       -- no intermediate header
        gre(3),          -- GRE encapsulation
        minimal(4),      -- Minimal encapsulation
        l2tp(5),         -- L2TP encapsulation
        pptp(6),         -- PPTP encapsulation
        l2f(7),          -- L2F encapsulation
        udp(8),          -- UDP encapsulation
        atmp(9),         -- ATMP encapsulation
        msdp(10),        -- MSDP encapsulation
        sixToFour(11),   -- 6to4 encapsulation
        sixOverFour(12), -- 6over4 encapsulation
        isatap(13),      -- ISATAP encapsulation
        teredo(14),      -- Teredo encapsulation
        ipHttps(15)      -- IPHTTPS
    }

END
```

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/lib/mibs/IANAifType-MIB.mib
No license file was found, but licenses were detected in source scan.

Network Working Group Editors of this version:
Request for Comments: 2578 K. McCloghrie
STD: 58 Cisco Systems
Obsoletes: 1902 D. Perkins
Category: Standards Track SNMPinfo
J. Schoenwaelder
TU Braunschweig
Authors of previous version:
J. Case
SNMP Research
K. McCloghrie
Cisco Systems
M. Rose
First Virtual Holdings
S. Waldbusser
International Network Services
April 1999

Structure of Management Information Version 2 (SMIv2)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Table of Contents

1. Introduction

Management information is viewed as a collection of managed objects, residing in a virtual information store, termed the Management Information Base (MIB). Collections of related objects are defined in MIB modules. These modules are written using an adapted subset of

OSI's Abstract Syntax Notation One, ASN.1 (1988) [1]. It is the purpose of this document, the Structure of Management Information (SMI), to define that adapted subset, and to assign a set of associated administrative values.

The SMI is divided into three parts: module definitions, object definitions, and, notification definitions.

(1) Module definitions are used when describing information modules. An ASN.1 macro, MODULE-IDENTITY, is used to concisely convey the semantics of an information module.

(2) Object definitions are used when describing managed objects. An ASN.1 macro, OBJECT-TYPE, is used to concisely convey the syntax and semantics of a managed object.

(3) Notification definitions are used when describing unsolicited transmissions of management information. An ASN.1 macro, NOTIFICATION-TYPE, is used to concisely convey the syntax and semantics of a notification.

1.1. A Note on Terminology

For the purpose of exposition, the original Structure of Management Information, as described in RFCs 1155 (STD 16), 1212 (STD 16), and RFC 1215, is termed the SMI version 1 (SMIv1). The current version of the Structure of Management Information is termed SMI version 2 (SMIv2).

2. Definitions

SNMPv2-SMI DEFINITIONS ::= BEGIN

-- the path to the root

```
org            OBJECT IDENTIFIER ::= { iso 3 }  --  "iso" = 1
dod            OBJECT IDENTIFIER ::= { org 6 }
internet       OBJECT IDENTIFIER ::= { dod 1 }

directory      OBJECT IDENTIFIER ::= { internet 1 }

mgmt           OBJECT IDENTIFIER ::= { internet 2 }
mib-2          OBJECT IDENTIFIER ::= { mgmt 1 }
transmission   OBJECT IDENTIFIER ::= { mib-2 10 }

experimental   OBJECT IDENTIFIER ::= { internet 3 }

private        OBJECT IDENTIFIER ::= { internet 4 }
enterprises    OBJECT IDENTIFIER ::= { private 1 }

security       OBJECT IDENTIFIER ::= { internet 5 }

snmpV2         OBJECT IDENTIFIER ::= { internet 6 }

-- transport domains
snmpDomains    OBJECT IDENTIFIER ::= { snmpV2 1 }

-- transport proxies
snmpProxys     OBJECT IDENTIFIER ::= { snmpV2 2 }

-- module identities
snmpModules    OBJECT IDENTIFIER ::= { snmpV2 3 }

-- Extended UTCTime, to allow dates with four-digit years
-- (Note that this definition of ExtUTCTime is not to be IMPORTed
--  by MIB modules.)
ExtUTCTime ::= OCTET STRING(SIZE(11 | 13))
  -- format is YYMMDDHHMMZ or YYYYMMDDHHMMZ
```

McCloghrie, et al.   Standards Track   [Page 4]

RFC 2578   SMIv2   April 1999

```
  --   where: YY   - last two digits of year (only years
  --                 between 1900-1999)
  --          YYYY - last four digits of the year (any year)
  --          MM   - month (01 through 12)
```

```
  --        DD   - day of month (01 through 31)
  --        HH   - hours (00 through 23)
  --        MM   - minutes (00 through 59)
  --        Z    - denotes GMT (the ASCII character Z)
  --
  -- For example, "9502192015Z" and "199502192015Z" represent
  -- 8:15pm GMT on 19 February 1995. Years after 1999 must use
  -- the four digit year format. Years 1900-1999 may use the
  -- two or four digit format.

-- definitions for information modules

MODULE-IDENTITY MACRO ::=
BEGIN
  TYPE NOTATION ::=
                "LAST-UPDATED" value(Update ExtUTCTime)
                "ORGANIZATION" Text
                "CONTACT-INFO" Text
                "DESCRIPTION" Text
                RevisionPart

  VALUE NOTATION ::=
                value(VALUE OBJECT IDENTIFIER)

  RevisionPart ::=
                Revisions
              | empty
  Revisions ::=
                Revision
              | Revisions Revision
  Revision ::=
                "REVISION" value(Update ExtUTCTime)
                "DESCRIPTION" Text

  -- a character string as defined in section 3.1.1
  Text ::= value(IA5String)
END


OBJECT-IDENTITY MACRO ::=
BEGIN
  TYPE NOTATION ::=
                "STATUS" Status
                "DESCRIPTION" Text
```

```
              ReferPart

   VALUE NOTATION ::=
              value(VALUE OBJECT IDENTIFIER)

   Status ::=
              "current"
            | "deprecated"
            | "obsolete"

   ReferPart ::=
              "REFERENCE" Text
            | empty

   -- a character string as defined in section 3.1.1
   Text ::= value(IA5String)
END


-- names of objects
-- (Note that these definitions of ObjectName and NotificationName
--  are not to be IMPORTed by MIB modules.)

ObjectName ::=
   OBJECT IDENTIFIER

NotificationName ::=
   OBJECT IDENTIFIER

-- syntax of objects

-- the "base types" defined here are:
--   3 built-in ASN.1 types: INTEGER, OCTET STRING, OBJECT IDENTIFIER
--   8 application-defined types: Integer32, IpAddress, Counter32,
--              Gauge32, Unsigned32, TimeTicks, Opaque, and Counter64

ObjectSyntax ::=
   CHOICE {
       simple
           SimpleSyntax,
```

```
        -- note that SEQUENCEs for conceptual tables and
        -- rows are not mentioned here...

      application-wide
          ApplicationSyntax
  }
```

McCloghrie, et al.   Standards Track   [Page 6]

RFC 2578   SMIv2   April 1999

```
-- built-in ASN.1 types

SimpleSyntax ::=
  CHOICE {
      -- INTEGERs with a more restrictive range
      -- may also be used
      integer-value               -- includes Integer32
          INTEGER (-2147483648..2147483647),

      -- OCTET STRINGs with a more restrictive size
      -- may also be used
      string-value
          OCTET STRING (SIZE (0..65535)),

      objectID-value
          OBJECT IDENTIFIER
  }

-- indistinguishable from INTEGER, but never needs more than
-- 32-bits for a two's complement representation
Integer32 ::=
      INTEGER (-2147483648..2147483647)


-- application-wide types

ApplicationSyntax ::=
  CHOICE {
      ipAddress-value
          IpAddress,
```

```
        counter-value
           Counter32,

        timeticks-value
           TimeTicks,

        arbitrary-value
           Opaque,

        big-counter-value
           Counter64,

        unsigned-integer-value  -- includes Gauge32
           Unsigned32
    }

-- in network-byte order
```

McCloghrie, et al.   Standards Track   [Page 7]

RFC 2578   SMIv2   April 1999

```
-- (this is a tagged type for historical reasons)
IpAddress ::=
    [APPLICATION 0]
        IMPLICIT OCTET STRING (SIZE (4))

-- this wraps
Counter32 ::=
    [APPLICATION 1]
        IMPLICIT INTEGER (0..4294967295)

-- this doesn't wrap
Gauge32 ::=
    [APPLICATION 2]
        IMPLICIT INTEGER (0..4294967295)

-- an unsigned 32-bit quantity
-- indistinguishable from Gauge32
Unsigned32 ::=
    [APPLICATION 2]
```

```
      IMPLICIT INTEGER (0..4294967295)

-- hundredths of seconds since an epoch
TimeTicks ::=
   [APPLICATION 3]
      IMPLICIT INTEGER (0..4294967295)

-- for backward-compatibility only
Opaque ::=
   [APPLICATION 4]
      IMPLICIT OCTET STRING

-- for counters that wrap in less than one hour with only 32 bits
Counter64 ::=
   [APPLICATION 6]
      IMPLICIT INTEGER (0..18446744073709551615)


-- definition for objects

OBJECT-TYPE MACRO ::=
BEGIN
   TYPE NOTATION ::=
               "SYNTAX" Syntax
               UnitsPart
               "MAX-ACCESS" Access
               "STATUS" Status
               "DESCRIPTION" Text
               ReferPart
```

McCloghrie, et al.   Standards Track   [Page 8]

RFC 2578   SMIv2   April 1999

```
               IndexPart
               DefValPart

   VALUE NOTATION ::=
               value(VALUE ObjectName)

   Syntax ::=   -- Must be one of the following:
                    -- a base type (or its refinement),
```

```
                -- a textual convention (or its refinement), or
                -- a BITS pseudo-type
              type
            | "BITS" "{" NamedBits "}"

    NamedBits ::= NamedBit
            | NamedBits "," NamedBit

    NamedBit ::=  identifier "(" number ")" -- number is nonnegative

    UnitsPart ::=
              "UNITS" Text
            | empty

    Access ::=
              "not-accessible"
            | "accessible-for-notify"
            | "read-only"
            | "read-write"
            | "read-create"

    Status ::=
              "current"
            | "deprecated"
            | "obsolete"

    ReferPart ::=
              "REFERENCE" Text
            | empty

    IndexPart ::=
              "INDEX"    "{" IndexTypes "}"
            | "AUGMENTS" "{" Entry      "}"
            | empty
    IndexTypes ::=
              IndexType
            | IndexTypes "," IndexType
    IndexType ::=
              "IMPLIED" Index
            | Index
```

```
  Index ::=
                -- use the SYNTAX value of the
                -- correspondent OBJECT-TYPE invocation
              value(ObjectName)
  Entry ::=
                -- use the INDEX value of the
                -- correspondent OBJECT-TYPE invocation
              value(ObjectName)

  DefValPart ::= "DEFVAL" "{" Defvalue "}"
            | empty

  Defvalue ::=  -- must be valid for the type specified in
              -- SYNTAX clause of same OBJECT-TYPE macro
              value(ObjectSyntax)
            | "{" BitsValue "}"

  BitsValue ::= BitNames
            | empty

  BitNames ::=  BitName
            | BitNames "," BitName

  BitName ::= identifier

  -- a character string as defined in section 3.1.1
  Text ::= value(IA5String)
END


-- definitions for notifications

NOTIFICATION-TYPE MACRO ::=
BEGIN
  TYPE NOTATION ::=
              ObjectsPart
              "STATUS" Status
              "DESCRIPTION" Text
              ReferPart

  VALUE NOTATION ::=
              value(VALUE NotificationName)

  ObjectsPart ::=
              "OBJECTS" "{" Objects "}"
```

```
             | empty
   Objects ::=
              Object
```

McCloghrie, et al. Standards Track [Page 10]


```
             | Objects "," Object
   Object ::=
              value(ObjectName)

   Status ::=
              "current"
             | "deprecated"
             | "obsolete"

   ReferPart ::=
              "REFERENCE" Text
             | empty

   -- a character string as defined in section 3.1.1
   Text ::= value(IA5String)
END

-- definitions of administrative identifiers

zeroDotZero    OBJECT-IDENTITY
   STATUS     current
   DESCRIPTION
          "A value used for null identifiers."
   ::= { 0 0 }

END
```

3. Information Modules

An "information module" is an ASN.1 module defining information relating to network management.

The SMI describes how to use an adapted subset of ASN.1 (1988) to define an information module. Further, additional restrictions are placed on "standard" information modules. It is strongly recommended that "enterprise-specific" information modules also adhere to these restrictions.

Typically, there are three kinds of information modules:

(1) MIB modules, which contain definitions of inter-related managed objects, make use of the OBJECT-TYPE and NOTIFICATION-TYPE macros;

(2) compliance statements for MIB modules, which make use of the MODULE-COMPLIANCE and OBJECT-GROUP macros [2]; and,

(3) capability statements for agent implementations which make use of the AGENT-CAPABILITIES macros [2].

This classification scheme does not imply a rigid taxonomy. For example, a "standard" information module will normally include definitions of managed objects and a compliance statement. Similarly, an "enterprise-specific" information module might include definitions of managed objects and a capability statement. Of course, a "standard" information module may not contain capability statements.

The constructs of ASN.1 allowed in SMIv2 information modules include: the IMPORTS clause, value definitions for OBJECT IDENTIFIERs, type definitions for SEQUENCEs (with restrictions), ASN.1 type assignments of the restricted ASN.1 types allowed in SMIv2, and instances of ASN.1 macros defined in this document and its companion documents [2, 3]. Additional ASN.1 macros must not be defined in SMIv2 information modules. SMIv1 macros must not be used in SMIv2 information modules.

The names of all standard information modules must be unique (but different versions of the same information module should have the same name). Developers of enterprise information modules are encouraged to choose names for their information modules that will have a low probability of colliding with standard or other enterprise information modules. An information module may not use the ASN.1 construct of placing an object identifier value between the module name and the "DEFINITIONS" keyword. For the purposes of this specification, an ASN.1 module name begins with an upper-case letter and continues with zero or more letters, digits, or hyphens, except that a hyphen can not be the last character, nor can there be two consecutive hyphens.

All information modules start with exactly one invocation of the MODULE-IDENTITY macro, which provides contact information as well as revision history to distinguish between versions of the same information module. This invocation must appear immediately after any IMPORTs statements.

3.1. Macro Invocation

Within an information module, each macro invocation appears as:

<descriptor> <macro> <clauses> ::= <value>

where <descriptor> corresponds to an ASN.1 identifier, <macro> names the macro being invoked, and <clauses> and <value> depend on the definition of the macro. (Note that this definition of a descriptor applies to all macros defined in this memo and in [2].)

For the purposes of this specification, an ASN.1 identifier consists of one or more letters or digits, and its initial character must be a lower-case letter. Note that hyphens are not allowed by this specification (except for use by information modules converted from SMIv1 which did allow hyphens).

For all descriptors appearing in an information module, the descriptor shall be unique and mnemonic, and shall not exceed 64 characters in length. (However, descriptors longer than 32 characters are not recommended.) This promotes a common language for humans to use when discussing the information module and also facilitates simple table mappings for user-interfaces.

The set of descriptors defined in all "standard" information modules shall be unique.

Finally, by convention, if the descriptor refers to an object with a SYNTAX clause value of either Counter32 or Counter64, then the descriptor used for the object should denote plurality.

3.1.1. Textual Values and Strings

Some clauses in a macro invocation may take a character string as a textual value (e.g., the DESCRIPTION clause). Other clauses take binary or hexadecimal strings (in any position where a non-negative number is allowed).

A character string is preceded and followed by the quote character ("), and consists of an arbitrary number (possibly zero) of:

- any 7-bit displayable ASCII characters except quote ("),
- tab characters,
- spaces, and
- line terminator characters (\n or \r\n).

The value of a character string is interpreted as ASCII.

A binary string consists of a number (possibly zero) of zeros and ones preceded by a single (') and followed by either the pair ('B) or ('b), where the number is a multiple of eight.

A hexadecimal string consists of an even number (possibly zero) of hexadecimal digits, preceded by a single (') and followed by either the pair ('H) or ('h). Digits specified via letters can be in upper or lower case.

Note that ASN.1 comments can not be enclosed inside any of these types of strings.

3.2. IMPORTing Symbols

To reference an external object, the IMPORTS statement must be used
to identify both the descriptor and the module in which the
descriptor is defined, where the module is identified by its ASN.1
module name.

Note that when symbols from "enterprise-specific" information modules
are referenced (e.g., a descriptor), there is the possibility of
collision. As such, if different objects with the same descriptor
are IMPORTed, then this ambiguity is resolved by prefixing the
descriptor with the name of the information module and a dot ("."),
i.e.,

"module.descriptor"

(All descriptors must be unique within any information module.)

Of course, this notation can be used to refer to objects even when
there is no collision when IMPORTing symbols.

Finally, if any of the ASN.1 named types and macros defined in this
document, specifically:

Counter32, Counter64, Gauge32, Integer32, IpAddress, MODULE-
IDENTITY, NOTIFICATION-TYPE, Opaque, OBJECT-TYPE, OBJECT-
IDENTITY, TimeTicks, Unsigned32,

or any of those defined in [2] or [3], are used in an information
module, then they must be imported using the IMPORTS statement.
However, the following must not be included in an IMPORTS statement:

- named types defined by ASN.1 itself, specifically: INTEGER,
  OCTET STRING, OBJECT IDENTIFIER, SEQUENCE, SEQUENCE OF type,
- the BITS construct.

3.3. Exporting Symbols

The ASN.1 EXPORTS statement is not allowed in SMIv2 information
modules. All items defined in an information module are
automatically exported.

3.4. ASN.1 Comments

ASN.1 comments can be included in an information module. However, it
is recommended that all substantive descriptions be placed within an
appropriate DESCRIPTION clause.

ASN.1 comments commence with a pair of adjacent hyphens and end with the next pair of adjacent hyphens or at the end of the line, whichever occurs first. Comments ended by a pair of hyphens have the effect of a single space character.

3.5. OBJECT IDENTIFIER values

An OBJECT IDENTIFIER value is an ordered list of non-negative numbers. For the SMIv2, each number in the list is referred to as a sub-identifier, there are at most 128 sub-identifiers in a value, and each sub-identifier has a maximum value of 2^32-1 (4294967295 decimal).

All OBJECT IDENTIFIER values have at least two sub-identifiers, where the value of the first sub-identifier is one of the following well-known names:

Value	Name
0	ccitt
1	iso
2	joint-iso-ccitt

(Note that this SMI does not recognize "new" well-known names, e.g., as defined when the CCITT became the ITU.)

3.6. OBJECT IDENTIFIER usage

OBJECT IDENTIFIERs are used in information modules in two ways:

(1) registration: the definition of a particular item is registered as a particular OBJECT IDENTIFIER value, and associated with a particular descriptor. After such a registration, the semantics thereby associated with the value are not allowed to change, the OBJECT IDENTIFIER can not be used for any other registration, and the descriptor can not be changed nor associated with any other registration. The following macros result in a registration:

OBJECT-TYPE, MODULE-IDENTITY, NOTIFICATION-TYPE, OBJECT-GROUP,

OBJECT-IDENTITY, NOTIFICATION-GROUP, MODULE-COMPLIANCE,
AGENT-CAPABILITIES.

(2) assignment: a descriptor can be assigned to a particular OBJECT IDENTIFIER value. For this usage, the semantics associated with the OBJECT IDENTIFIER value is not allowed to change, and a descriptor assigned to a particular OBJECT IDENTIFIER value cannot subsequently be assigned to another. However, multiple descriptors can be assigned to the same OBJECT IDENTIFIER value. Such assignments are specified in the following manner:


```
mib         OBJECT IDENTIFIER ::= { mgmt 1 }  -- from RFC1156
mib-2       OBJECT IDENTIFIER ::= { mgmt 1 }  -- from RFC1213
fredRouter  OBJECT IDENTIFIER ::= { flintStones 1 1 }
barneySwitch OBJECT IDENTIFIER ::= { flintStones bedrock(2) 1 }
```

Note while the above examples are legal, the following is not:

```
dinoHost OBJECT IDENTIFIER ::= { flintStones bedrock 2 }
```

A descriptor is allowed to be associated with both a registration and an assignment, providing both are associated with the same OBJECT IDENTIFIER value and semantics.

3.7. Reserved Keywords

The following are reserved keywords which must not be used as descriptors or module names:

ABSENT ACCESS AGENT-CAPABILITIES ANY APPLICATION AUGMENTS BEGIN
BIT BITS BOOLEAN BY CHOICE COMPONENT COMPONENTS CONTACT-INFO
CREATION-REQUIRES Counter32 Counter64 DEFAULT DEFINED
DEFINITIONS DEFVAL DESCRIPTION DISPLAY-HINT END ENUMERATED
ENTERPRISE EXPLICIT EXPORTS EXTERNAL FALSE FROM GROUP Gauge32
IDENTIFIER IMPLICIT IMPLIED IMPORTS INCLUDES INDEX INTEGER
Integer32 IpAddress LAST-UPDATED MANDATORY-GROUPS MAX MAX-ACCESS
MIN MIN-ACCESS MINUS-INFINITY MODULE MODULE-COMPLIANCE MODULE-
IDENTITY NOTIFICATION-GROUP NOTIFICATION-TYPE NOTIFICATIONS NULL

OBJECT OBJECT-GROUP OBJECT-IDENTITY OBJECT-TYPE OBJECTS OCTET OF OPTIONAL ORGANIZATION Opaque PLUS-INFINITY PRESENT PRIVATE PRODUCT-RELEASE REAL REFERENCE REVISION SEQUENCE SET SIZE STATUS STRING SUPPORTS SYNTAX TAGS TEXTUAL-CONVENTION TRAP-TYPE TRUE TimeTicks UNITS UNIVERSAL Unsigned32 VARIABLES VARIATION WITH WRITE-SYNTAX

4. Naming Hierarchy

The root of the subtree administered by the Internet Assigned Numbers Authority (IANA) for the Internet is:

internet OBJECT IDENTIFIER ::= { iso 3 6 1 }

That is, the Internet subtree of OBJECT IDENTIFIERs starts with the prefix:

1.3.6.1.

Several branches underneath this subtree are used for network management:


```
mgmt          OBJECT IDENTIFIER ::= { internet 2 }
experimental  OBJECT IDENTIFIER ::= { internet 3 }
private       OBJECT IDENTIFIER ::= { internet 4 }
enterprises   OBJECT IDENTIFIER ::= { private 1 }
```

However, the SMI does not prohibit the definition of objects in other portions of the object tree.

The mgmt(2) subtree is used to identify "standard" objects.

The experimental(3) subtree is used to identify objects being designed by working groups of the IETF. If an information module produced by a working group becomes a "standard" information module, then at the very beginning of its entry onto the Internet standards track, the objects are moved under the mgmt(2) subtree.

The private(4) subtree is used to identify objects defined unilaterally. The enterprises(1) subtree beneath private is used, among other things, to permit providers of networking subsystems to register models of their products.

5. Mapping of the MODULE-IDENTITY macro

The MODULE-IDENTITY macro is used to provide contact and revision history for each information module. It must appear exactly once in every information module. It should be noted that the expansion of the MODULE-IDENTITY macro is something which conceptually happens during implementation and not during run-time.

Note that reference in an IMPORTS clause or in clauses of SMIv2 macros to an information module is NOT through the use of the 'descriptor' of a MODULE-IDENTITY macro; rather, an information module is referenced through specifying its module name.

5.1. Mapping of the LAST-UPDATED clause

The LAST-UPDATED clause, which must be present, contains the date and time that this information module was last edited.

5.2. Mapping of the ORGANIZATION clause

The ORGANIZATION clause, which must be present, contains a textual description of the organization under whose auspices this information module was developed.

5.3. Mapping of the CONTACT-INFO clause

The CONTACT-INFO clause, which must be present, contains the name, postal address, telephone number, and electronic mail address of the person to whom technical queries concerning this information module should be sent.

5.4. Mapping of the DESCRIPTION clause

The DESCRIPTION clause, which must be present, contains a high-level textual description of the contents of this information module.

5.5. Mapping of the REVISION clause

The REVISION clause, which need not be present, is repeatedly used to describe the revisions (including the initial version) made to this information module, in reverse chronological order (i.e., most recent first). Each instance of this clause contains the date and time of the revision.

5.5.1. Mapping of the DESCRIPTION sub-clause

The DESCRIPTION sub-clause, which must be present for each REVISION clause, contains a high-level textual description of the revision identified in that REVISION clause.

5.6. Mapping of the MODULE-IDENTITY value

The value of an invocation of the MODULE-IDENTITY macro is an OBJECT IDENTIFIER. As such, this value may be authoritatively used when specifying an OBJECT IDENTIFIER value to refer to the information module containing the invocation.

Note that it is a common practice to use the value of the MODULE-IDENTITY macro as a subtree under which other OBJECT IDENTIFIER values assigned within the module are defined. However, it is legal (and occasionally necessary) for the other OBJECT IDENTIFIER values assigned within the module to be unrelated to the OBJECT IDENTIFIER value of the MODULE-IDENTITY macro.

5.7. Usage Example

Consider how a skeletal MIB module might be constructed: e.g.,

```
FIZBIN-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, experimental
```

```
     FROM SNMPv2-SMI;

 fizbin MODULE-IDENTITY
    LAST-UPDATED "199505241811Z"
    ORGANIZATION "IETF SNMPv2 Working Group"
    CONTACT-INFO
            "       Marshall T. Rose

             Postal: Dover Beach Consulting, Inc.
                     420 Whisman Court
                     Mountain View, CA  94043-2186
                     US

                Tel: +1 415 968 1052
                Fax: +1 415 968 2510

             E-mail: mrose@dbc.mtview.ca.us"

    DESCRIPTION
            "The MIB module for entities implementing the xxxx
            protocol."
    REVISION      "9505241811Z"
    DESCRIPTION
            "The latest version of this MIB module."
    REVISION      "9210070433Z"
    DESCRIPTION
            "The initial version of this MIB module, published in
            RFC yyyy."
 -- contact IANA for actual number
    ::= { experimental xx }

 END
```

6. Mapping of the OBJECT-IDENTITY macro

The OBJECT-IDENTITY macro is used to define information about an OBJECT IDENTIFIER assignment. All administrative OBJECT IDENTIFIER assignments which define a type identification value (see AutonomousType, a textual convention defined in [3]) should be defined via the OBJECT-IDENTITY macro. It should be noted that the expansion of the OBJECT-IDENTITY macro is something which conceptually happens during implementation and not during run-time.

6.1. Mapping of the STATUS clause

The STATUS clause, which must be present, indicates whether this definition is current or historic.

The value "current" means that the definition is current and valid. The value "obsolete" means the definition is obsolete and should not be implemented and/or can be removed if previously implemented. While the value "deprecated" also indicates an obsolete definition, it permits new/continued implementation in order to foster interoperability with older/existing implementations.

6.2. Mapping of the DESCRIPTION clause

The DESCRIPTION clause, which must be present, contains a textual description of the object assignment.

6.3. Mapping of the REFERENCE clause

The REFERENCE clause, which need not be present, contains a textual cross-reference to some other document, either another information module which defines a related assignment, or some other document which provides additional information relevant to this definition.

6.4. Mapping of the OBJECT-IDENTITY value

The value of an invocation of the OBJECT-IDENTITY macro is an OBJECT IDENTIFIER.

6.5. Usage Example

Consider how an OBJECT IDENTIFIER assignment might be made: e.g.,

```
fizbin69 OBJECT-IDENTITY
    STATUS  current
    DESCRIPTION
```

"The authoritative identity of the Fizbin 69 chipset."
::= { fizbinChipSets 1 }

7. Mapping of the OBJECT-TYPE macro

The OBJECT-TYPE macro is used to define a type of managed object. It should be noted that the expansion of the OBJECT-TYPE macro is something which conceptually happens during implementation and not during run-time.

For leaf objects which are not columnar objects (i.e., not contained within a conceptual table), instances of the object are identified by appending a sub-identifier of zero to the name of that object. Otherwise, the INDEX clause of the conceptual row object superior to a columnar object defines instance identification information.

7.1. Mapping of the SYNTAX clause

The SYNTAX clause, which must be present, defines the abstract data structure corresponding to that object. The data structure must be one of the following: a base type, the BITS construct, or a textual convention. (SEQUENCE OF and SEQUENCE are also possible for conceptual tables, see section 7.1.12). The base types are those defined in the ObjectSyntax CHOICE. A textual convention is a newly-defined type defined as a sub-type of a base type [3].

An extended subset of the full capabilities of ASN.1 (1988) sub-typing is allowed, as appropriate to the underlying ASN.1 type. Any such restriction on size, range or enumerations specified in this clause represents the maximal level of support which makes "protocol sense". Restrictions on sub-typing are specified in detail in Section 9 and Appendix A of this memo.

The semantics of ObjectSyntax are now described.

7.1.1. Integer32 and INTEGER

The Integer32 type represents integer-valued information between -2^31 and 2^31-1 inclusive (-2147483648 to 2147483647 decimal). This type is indistinguishable from the INTEGER type. Both the INTEGER and Integer32 types may be sub-typed to be more constrained than the Integer32 type.

The INTEGER type (but not the Integer32 type) may also be used to represent integer-valued information as named-number enumerations. In this case, only those named-numbers so enumerated may be present as a value. Note that although it is recommended that enumerated values start at 1 and be numbered contiguously, any valid value for Integer32 is allowed for an enumerated value and, further, enumerated values needn't be contiguously assigned.

Finally, a label for a named-number enumeration must consist of one or more letters or digits, up to a maximum of 64 characters, and the initial character must be a lower-case letter. (However, labels longer than 32 characters are not recommended.) Note that hyphens are not allowed by this specification (except for use by information modules converted from SMIv1 which did allow hyphens).

7.1.2. OCTET STRING

The OCTET STRING type represents arbitrary binary or textual data. Although the SMI-specified size limitation for this type is 65535 octets, MIB designers should realize that there may be implementation and interoperability limitations for sizes in excess of 255 octets.

7.1.3. OBJECT IDENTIFIER

The OBJECT IDENTIFIER type represents administratively assigned names. Any instance of this type may have at most 128 sub-identifiers. Further, each sub-identifier must not exceed the value 2^32-1 (4294967295 decimal).

7.1.4. The BITS construct

The BITS construct represents an enumeration of named bits. This collection is assigned non-negative, contiguous (but see below) values, starting at zero. Only those named-bits so enumerated may be present in a value. (Thus, enumerations must be assigned to consecutive bits; however, see Section 9 for refinements of an object with this syntax.)

As part of updating an information module, for an object defined using the BITS construct, new enumerations can be added or existing enumerations can have new labels assigned to them. After an enumeration is added, it might not be possible to distinguish between an implementation of the updated object for which the new enumeration is not asserted, and an implementation of the object prior to the addition. Depending on the circumstances, such an ambiguity could either be desirable or could be undesirable. The means to avoid such an ambiguity is dependent on the encoding of values on the wire; however, one possibility is to define new enumerations starting at the next multiple of eight bits. (Of course, this can also result in the enumerations no longer being contiguous.)

Although there is no SMI-specified limitation on the number of enumerations (and therefore on the length of a value), except as may be imposed by the limit on the length of an OCTET STRING, MIB designers should realize that there may be implementation and interoperability limitations for sizes in excess of 128 bits.

Finally, a label for a named-number enumeration must consist of one or more letters or digits, up to a maximum of 64 characters, and the initial character must be a lower-case letter. (However, labels longer than 32 characters are not recommended.) Note that hyphens are not allowed by this specification.

7.1.5. IpAddress

The IpAddress type represents a 32-bit internet address. It is represented as an OCTET STRING of length 4, in network byte-order.

Note that the IpAddress type is a tagged type for historical reasons. Network addresses should be represented using an invocation of the TEXTUAL-CONVENTION macro [3].

7.1.6. Counter32

The Counter32 type represents a non-negative integer which monotonically increases until it reaches a maximum value of 2^32-1 (4294967295 decimal), when it wraps around and starts increasing again from zero.

Counters have no defined "initial" value, and thus, a single value of a Counter has (in general) no information content. Discontinuities in the monotonically increasing value normally occur at re-initialization of the management system, and at other times as specified in the description of an object-type using this ASN.1 type. If such other times can occur, for example, the creation of an object instance at times other than re-initialization, then a corresponding object should be defined, with an appropriate SYNTAX clause, to indicate the last discontinuity. Examples of appropriate SYNTAX clause include: TimeStamp (a textual convention defined in [3]), DateAndTime (another textual convention from [3]) or TimeTicks.

The value of the MAX-ACCESS clause for objects with a SYNTAX clause value of Counter32 is either "read-only" or "accessible-for-notify".

A DEFVAL clause is not allowed for objects with a SYNTAX clause value of Counter32.

7.1.7. Gauge32

The Gauge32 type represents a non-negative integer, which may increase or decrease, but shall never exceed a maximum value, nor fall below a minimum value. The maximum value can not be greater than 2^32-1 (4294967295 decimal), and the minimum value can not be smaller than 0. The value of a Gauge32 has its maximum value whenever the information being modeled is greater than or equal to its maximum value, and has its minimum value whenever the information being modeled is smaller than or equal to its minimum value. If the information being modeled subsequently decreases below (increases above) the maximum (minimum) value, the Gauge32 also decreases (increases). (Note that despite of the use of the term "latched" in the original definition of this type, it does not become "stuck" at its maximum or minimum value.)

7.1.8. TimeTicks

The TimeTicks type represents a non-negative integer which represents the time, modulo 2^32 (4294967296 decimal), in hundredths of a second between two epochs. When objects are defined which use this ASN.1 type, the description of the object identifies both of the reference epochs.

For example, [3] defines the TimeStamp textual convention which is based on the TimeTicks type. With a TimeStamp, the first reference epoch is defined as the time when sysUpTime [5] was zero, and the second reference epoch is defined as the current value of sysUpTime.

The TimeTicks type may not be sub-typed.

7.1.9. Opaque

The Opaque type is provided solely for backward-compatibility, and shall not be used for newly-defined object types.

The Opaque type supports the capability to pass arbitrary ASN.1 syntax. A value is encoded using the ASN.1 Basic Encoding Rules [4] into a string of octets. This, in turn, is encoded as an OCTET STRING, in effect "double-wrapping" the original ASN.1 value.

Note that a conforming implementation need only be able to accept and recognize opaquely-encoded data. It need not be able to unwrap the data and then interpret its contents.

A requirement on "standard" MIB modules is that no object may have a SYNTAX clause value of Opaque.

7.1.10. Counter64

The Counter64 type represents a non-negative integer which monotonically increases until it reaches a maximum value of 2^64-1 (18446744073709551615 decimal), when it wraps around and starts increasing again from zero.

Counters have no defined "initial" value, and thus, a single value of a Counter has (in general) no information content. Discontinuities in the monotonically increasing value normally occur at re-initialization of the management system, and at other times as specified in the description of an object-type using this ASN.1 type. If such other times can occur, for example, the creation of an object instance at times other than re-initialization, then a corresponding object should be defined, with an appropriate SYNTAX clause, to indicate the last discontinuity. Examples of appropriate SYNTAX

clause are: TimeStamp (a textual convention defined in [3]), DateAndTime (another textual convention from [3]) or TimeTicks.

The value of the MAX-ACCESS clause for objects with a SYNTAX clause value of Counter64 is either "read-only" or "accessible-for-notify".

A requirement on "standard" MIB modules is that the Counter64 type may be used only if the information being modeled would wrap in less than one hour if the Counter32 type was used instead.

A DEFVAL clause is not allowed for objects with a SYNTAX clause value of Counter64.

7.1.11. Unsigned32

The Unsigned32 type represents integer-valued information between 0 and 2^32-1 inclusive (0 to 4294967295 decimal).

7.1.12. Conceptual Tables

Management operations apply exclusively to scalar objects. However, it is sometimes convenient for developers of management applications to impose an imaginary, tabular structure on an ordered collection of objects within the MIB. Each such conceptual table contains zero or more rows, and each row may contain one or more scalar objects, termed columnar objects. This conceptualization is formalized by using the OBJECT-TYPE macro to define both an object which corresponds to a table and an object which corresponds to a row in that table. A conceptual table has SYNTAX of the form:

SEQUENCE OF <EntryType>

where <EntryType> refers to the SEQUENCE type of its subordinate conceptual row. A conceptual row has SYNTAX of the form:

<EntryType>

where <EntryType> is a SEQUENCE type defined as follows:

<EntryType> ::= SEQUENCE { <type1>, ... , <typeN> }

where there is one <type> for each subordinate object, and each <type> is of the form:

<descriptor> <syntax>

where <descriptor> is the descriptor naming a subordinate object, and <syntax> has the value of that subordinate object's SYNTAX clause,

except that both sub-typing information and the named values for enumerated integers or the named bits for the BITS construct, are omitted from <syntax>.

Further, a <type> is always present for every subordinate object. (The ASN.1 DEFAULT and OPTIONAL clauses are disallowed in the SEQUENCE definition.) The MAX-ACCESS clause for conceptual tables and rows is "not-accessible".

7.1.12.1. Creation and Deletion of Conceptual Rows

For newly-defined conceptual rows which allow the creation of new object instances and/or the deletion of existing object instances, there should be one columnar object with a SYNTAX clause value of RowStatus (a textual convention defined in [3]) and a MAX-ACCESS clause value of read-create. By convention, this is termed the status column for the conceptual row.

7.2. Mapping of the UNITS clause

This UNITS clause, which need not be present, contains a textual definition of the units associated with that object.

7.3. Mapping of the MAX-ACCESS clause

The MAX-ACCESS clause, which must be present, defines whether it makes "protocol sense" to read, write and/or create an instance of the object, or to include its value in a notification. This is the maximal level of access for the object. (This maximal level of access is independent of any administrative authorization policy.)

The value "read-write" indicates that read and write access make "protocol sense", but create does not. The value "read-create" indicates that read, write and create access make "protocol sense". The value "not-accessible" indicates an auxiliary object (see Section 7.7). The value "accessible-for-notify" indicates an object which is accessible only via a notification (e.g., snmpTrapOID [5]).

These values are ordered, from least to greatest: "not-accessible", "accessible-for-notify", "read-only", "read-write", "read-create".

If any columnar object in a conceptual row has "read-create" as its maximal level of access, then no other columnar object of the same conceptual row may have a maximal access of "read-write". (Note that "read-create" is a superset of "read-write".)

7.4. Mapping of the STATUS clause

The STATUS clause, which must be present, indicates whether this definition is current or historic.

The value "current" means that the definition is current and valid. The value "obsolete" means the definition is obsolete and should not be implemented and/or can be removed if previously implemented. While the value "deprecated" also indicates an obsolete definition, it permits new/continued implementation in order to foster interoperability with older/existing implementations.

7.5. Mapping of the DESCRIPTION clause

The DESCRIPTION clause, which must be present, contains a textual definition of that object which provides all semantic definitions necessary for implementation, and should embody any information which would otherwise be communicated in any ASN.1 commentary annotations associated with the object.

7.6. Mapping of the REFERENCE clause

The REFERENCE clause, which need not be present, contains a textual cross-reference to some other document, either another information module which defines a related assignment, or some other document which provides additional information relevant to this definition.

7.7. Mapping of the INDEX clause

The INDEX clause, which must be present if that object corresponds to a conceptual row (unless an AUGMENTS clause is present instead), and must be absent otherwise, defines instance identification information for the columnar objects subordinate to that object.

The instance identification information in an INDEX clause must specify object(s) such that value(s) of those object(s) will unambiguously distinguish a conceptual row. The objects can be columnar objects from the same and/or another conceptual table, but must not be scalar objects. Multiple occurrences of the same object in a single INDEX clause is strongly discouraged.

The syntax of the objects in the INDEX clause indicate how to form the instance-identifier:

(1) integer-valued (i.e., having INTEGER as its underlying primitive type): a single sub-identifier taking the integer value (this works only for non-negative integers);

(2) string-valued, fixed-length strings (or variable-length preceded by the IMPLIED keyword): `n' sub-identifiers, where `n' is the length of the string (each octet of the string is encoded in a separate sub-identifier);

(3) string-valued, variable-length strings (not preceded by the IMPLIED keyword): `n+1' sub-identifiers, where `n' is the length of the string (the first sub-identifier is `n' itself, following this, each octet of the string is encoded in a separate sub-identifier);

(4) object identifier-valued (when preceded by the IMPLIED keyword): `n' sub-identifiers, where `n' is the number of sub-identifiers in the value (each sub-identifier of the value is copied into a separate sub-identifier);

(5) object identifier-valued (when not preceded by the IMPLIED keyword): `n+1' sub-identifiers, where `n' is the number of sub-identifiers in the value (the first sub-identifier is `n' itself, following this, each sub-identifier in the value is copied);

(6) IpAddress-valued: 4 sub-identifiers, in the familiar a.b.c.d notation.

Note that the IMPLIED keyword can only be present for an object having a variable-length syntax (e.g., variable-length strings or object identifier-valued objects), Further, the IMPLIED keyword can only be associated with the last object in the INDEX clause. Finally, the IMPLIED keyword may not be used on a variable-length string object if that string might have a value of zero-length.

Since a single value of a Counter has (in general) no information content (see section 7.1.6 and 7.1.10), objects defined using the syntax, Counter32 or Counter64, must not be specified in an INDEX clause. If an object defined using the BITS construct is used in an INDEX clause, it is considered a variable-length string.

Instances identified by use of integer-valued objects should be numbered starting from one (i.e., not from zero). The use of zero as a value for an integer-valued index object should be avoided, except in special cases.

Objects which are both specified in the INDEX clause of a conceptual row and also columnar objects of the same conceptual row are termed auxiliary objects. The MAX-ACCESS clause for auxiliary objects is "not-accessible", except in the following circumstances:

(1) within a MIB module originally written to conform to SMIv1, and later converted to conform to SMIv2; or

(2) a conceptual row must contain at least one columnar object which is not an auxiliary object. In the event that all of a conceptual row's columnar objects are also specified in its INDEX clause, then one of them must be accessible, i.e., have a MAX-ACCESS clause of "read-only". (Note that this situation does not arise for a conceptual row allowing create access, since such a row will have a status column which will not be an auxiliary object.)

Note that objects specified in a conceptual row's INDEX clause need not be columnar objects of that conceptual row. In this situation, the DESCRIPTION clause of the conceptual row must include a textual explanation of how the objects which are included in the INDEX clause but not columnar objects of that conceptual row, are used in uniquely identifying instances of the conceptual row's columnar objects.

7.8. Mapping of the AUGMENTS clause

The AUGMENTS clause, which must not be present unless the object corresponds to a conceptual row, is an alternative to the INDEX clause. Every object corresponding to a conceptual row has either an INDEX clause or an AUGMENTS clause.

If an object corresponding to a conceptual row has an INDEX clause, that row is termed a base conceptual row; alternatively, if the object has an AUGMENTS clause, the row is said to be a conceptual row augmentation, where the AUGMENTS clause names the object corresponding to the base conceptual row which is augmented by this conceptual row augmentation. (Thus, a conceptual row augmentation cannot itself be augmented.) Instances of subordinate columnar objects of a conceptual row augmentation are identified according to the INDEX clause of the base conceptual row corresponding to the object named in the AUGMENTS clause. Further, instances of subordinate columnar objects of a conceptual row augmentation exist according to the same semantics as instances of subordinate columnar objects of the base conceptual row being augmented. As such, note that creation of a base conceptual row implies the correspondent creation of any conceptual row augmentations.

For example, a MIB designer might wish to define additional columns in an "enterprise-specific" MIB which logically extend a conceptual row in a "standard" MIB. The "standard" MIB definition of the conceptual row would include the INDEX clause and the "enterprise-specific" MIB would contain the definition of a conceptual row using the AUGMENTS clause. On the other hand, it would be incorrect to use the AUGMENTS clause for the relationship between RFC 2233's ifTable

and the many media-specific MIBs which extend it for specific media (e.g., the dot3Table in RFC 2358), since not all interfaces are of the same media.

Note that a base conceptual row may be augmented by multiple conceptual row augmentations.

7.8.1. Relation between INDEX and AUGMENTS clauses

When defining instance identification information for a conceptual table:

(1) If there is a one-to-one correspondence between the conceptual rows of this table and an existing table, then the AUGMENTS clause should be used.

(2) Otherwise, if there is a sparse relationship between the conceptual

rows of this table and an existing table, then an INDEX clause
should be used which is identical to that in the existing table.
For example, the relationship between RFC 2233's ifTable and a
media-specific MIB which extends the ifTable for a specific media
(e.g., the dot3Table in RFC 2358), is a sparse relationship.

(3) Otherwise, if no existing objects have the required syntax and
semantics, then auxiliary objects should be defined within the
conceptual row for the new table, and those objects should be used
within the INDEX clause for the conceptual row.

7.9. Mapping of the DEFVAL clause

The DEFVAL clause, which need not be present, defines an acceptable
default value which may be used at the discretion of an agent when an
object instance is created. That is, the value is a "hint" to
implementors.

During conceptual row creation, if an instance of a columnar object
is not present as one of the operands in the correspondent management
protocol set operation, then the value of the DEFVAL clause, if
present, indicates an acceptable default value that an agent might
use (especially for a read-only object).

Note that with this definition of the DEFVAL clause, it is
appropriate to use it for any columnar object of a read-create table.
It is also permitted to use it for scalar objects dynamically created
by an agent, or for columnar objects of a read-write table
dynamically created by an agent.

The value of the DEFVAL clause must, of course, correspond to the
SYNTAX clause for the object. If the value is an OBJECT IDENTIFIER,
then it must be expressed as a single ASN.1 identifier, and not as a
collection of sub-identifiers.

Note that if an operand to the management protocol set operation is an instance of a read-only object, then the error `notWritable' [6] will be returned. As such, the DEFVAL clause can be used to provide an acceptable default value that an agent might use.

By way of example, consider the following possible DEFVAL clauses:

```
ObjectSyntax       DEFVAL clause
----------------   ------------
Integer32          DEFVAL { 1 }
                   -- same for Gauge32, TimeTicks, Unsigned32
INTEGER            DEFVAL { valid } -- enumerated value
OCTET STRING       DEFVAL { 'ffffffffffff'H }
DisplayString      DEFVAL { "SNMP agent" }
IpAddress          DEFVAL { 'c0210415'H } -- 192.33.4.21
OBJECT IDENTIFIER  DEFVAL { sysDescr }
BITS               DEFVAL { { primary, secondary } }
                   -- enumerated values that are set
BITS               DEFVAL { { } }
                   -- no enumerated values are set
```

A binary string used in a DEFVAL clause for an OCTET STRING must be either an integral multiple of eight or zero bits in length; similarly, a hexadecimal string must be an even number of hexadecimal digits. The value of a character string used in a DEFVAL clause must not contain tab characters or line terminator characters.

Object types with SYNTAX of Counter32 and Counter64 may not have DEFVAL clauses, since they do not have defined initial values. However, it is recommended that they be initialized to zero.

7.10. Mapping of the OBJECT-TYPE value

The value of an invocation of the OBJECT-TYPE macro is the name of the object, which is an OBJECT IDENTIFIER, an administratively assigned name.

When an OBJECT IDENTIFIER is assigned to an object:

(1) If the object corresponds to a conceptual table, then only a single assignment, that for a conceptual row, is present immediately beneath that object. The administratively assigned name for the conceptual row object is derived by appending a sub-identifier of

"1" to the administratively assigned name for the conceptual table.

(2) If the object corresponds to a conceptual row, then at least one assignment, one for each column in the conceptual row, is present beneath that object. The administratively assigned name for each column is derived by appending a unique, positive sub-identifier to the administratively assigned name for the conceptual row.

(3) Otherwise, no other OBJECT IDENTIFIERs which are subordinate to the object may be assigned.

Note that the final sub-identifier of any administratively assigned name for an object shall be positive. A zero-valued final sub-identifier is reserved for future use.

7.11. Usage Example

Consider how one might define a conceptual table and its subordinates. (This example uses the RowStatus textual convention defined in [3].)

```
evalSlot OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The index number of the first unassigned entry in the
            evaluation table, or the value of zero indicating that
            all entries are assigned.

            A management station should create new entries in the
            evaluation table using this algorithm:  first, issue a
            management protocol retrieval operation to determine the
            value of evalSlot; and, second, issue a management
            protocol set operation to create an instance of the
            evalStatus object setting its value to createAndGo(4) or
            createAndWait(5).  If this latter operation succeeds,
            then the management station may continue modifying the
            instances corresponding to the newly created conceptual
            row, without fear of collision with other management
            stations."
   ::= { eval 1 }
```

```
evalTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EvalEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION


McCloghrie, et al.          Standards Track                    [Page 32]



RFC 2578                         SMIv2                        April 1999


            "The (conceptual) evaluation table."
    ::= { eval 2 }

evalEntry OBJECT-TYPE
    SYNTAX      EvalEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry (conceptual row) in the evaluation table."
    INDEX   { evalIndex }
    ::= { evalTable 1 }

EvalEntry ::=
    SEQUENCE {
        evalIndex       Integer32,
        evalString      DisplayString,
        evalValue       Integer32,
        evalStatus      RowStatus
    }

evalIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "The auxiliary variable used for identifying instances of
            the columnar objects in the evaluation table."
        ::= { evalEntry 1 }

evalString OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-create
```

```
    STATUS      current
    DESCRIPTION
            "The string to evaluate."
        ::= { evalEntry 2 }

  evalValue OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The value when evalString was last evaluated, or zero if
             no such value is available."
    DEFVAL  { 0 }
        ::= { evalEntry 3 }

  evalStatus OBJECT-TYPE
```



```
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
            "The status column used for creating, modifying, and
            deleting instances of the columnar objects in the
            evaluation table."
   DEFVAL  { active }
       ::= { evalEntry 4 }
```

8. Mapping of the NOTIFICATION-TYPE macro

The NOTIFICATION-TYPE macro is used to define the information contained within an unsolicited transmission of management information (i.e., within either a SNMPv2-Trap-PDU or InformRequest-PDU). It should be noted that the expansion of the NOTIFICATION-TYPE macro is something which conceptually happens during implementation and not during run-time.

8.1. Mapping of the OBJECTS clause

The OBJECTS clause, which need not be present, defines an ordered sequence of MIB object types. One and only one object instance for each occurrence of each object type must be present, and in the specified order, in every instance of the notification. If the same object type occurs multiple times in a notification's ordered sequence, then an object instance is present for each of them. An object type specified in this clause must not have an MAX-ACCESS clause of "not-accessible". The notification's DESCRIPTION clause must specify the information/meaning conveyed by each occurrence of each object type in the sequence. The DESCRIPTION clause must also specify which object instance is present for each object type in the notification.

Note that an agent is allowed, at its own discretion, to append as many additional objects as it considers useful to the end of the notification (i.e., after the objects defined by the OBJECTS clause).

8.2. Mapping of the STATUS clause

The STATUS clause, which must be present, indicates whether this definition is current or historic.

The value "current" means that the definition is current and valid. The value "obsolete" means the definition is obsolete and should not be implemented and/or can be removed if previously implemented. While the value "deprecated" also indicates an obsolete definition, it permits new/continued implementation in order to foster

interoperability with older/existing implementations.

8.3. Mapping of the DESCRIPTION clause

The DESCRIPTION clause, which must be present, contains a textual definition of the notification which provides all semantic definitions necessary for implementation, and should embody any information which would otherwise be communicated in any ASN.1 commentary annotations associated with the notification. In particular, the DESCRIPTION clause should document which instances of the objects mentioned in the OBJECTS clause should be contained within notifications of this type.

8.4. Mapping of the REFERENCE clause

The REFERENCE clause, which need not be present, contains a textual cross-reference to some other document, either another information module which defines a related assignment, or some other document which provides additional information relevant to this definition.

8.5. Mapping of the NOTIFICATION-TYPE value

The value of an invocation of the NOTIFICATION-TYPE macro is the name of the notification, which is an OBJECT IDENTIFIER, an administratively assigned name. In order to achieve compatibility with SNMPv1 traps, both when converting SMIv1 information modules to/from this SMI, and in the procedures employed by multi-lingual systems and proxy forwarding applications, the next to last sub-identifier in the name of any newly-defined notification must have the value zero.

Sections 4.2.6 and 4.2.7 of [6] describe how the NOTIFICATION-TYPE macro is used to generate a SNMPv2-Trap-PDU or InformRequest-PDU, respectively.

8.6. Usage Example

Consider how a configuration change notification might be described:

```
entityMIBTraps      OBJECT IDENTIFIER ::= { entityMIB 2 }
entityMIBTrapPrefix OBJECT IDENTIFIER ::= { entityMIBTraps 0 }

entConfigChange NOTIFICATION-TYPE
    STATUS          current
    DESCRIPTION
            "An entConfigChange trap is sent when the value of
            entLastChangeTime changes. It can be utilized by an NMS to
            trigger logical/physical entity table maintenance polls.
```

An agent must not generate more than one entConfigChange
'trap-event' in a five second period, where a 'trap-event'
is the transmission of a single trap PDU to a list of
trap destinations. If additional configuration changes
occur within the five second 'throttling' period, then
these trap-events should be suppressed by the agent. An
NMS should periodically check the value of
entLastChangeTime to detect any missed entConfigChange
trap-events, e.g. due to throttling or transmission loss."
::= { entityMIBTrapPrefix 1 }

According to this invocation, the notification authoritatively identified as

{ entityMIBTrapPrefix 1 }

is used to report a particular type of configuration change.

9. Refined Syntax

Some macros have clauses which allows syntax to be refined, specifically: the SYNTAX clause of the OBJECT-TYPE macro, and the SYNTAX/WRITE-SYNTAX clauses of the MODULE-COMPLIANCE and AGENT-CAPABILITIES macros [2]. However, not all refinements of syntax are appropriate. In particular, the object's primitive or application type must not be changed.

Further, the following restrictions apply:

Restrictions to Refinement of

object syntax	range	enumeration	size
INTEGER	(1)	(2)	-
Integer32	(1)	-	-
Unsigned32	(1)	-	-
OCTET STRING	-	-	(3)
OBJECT IDENTIFIER	-	-	-
BITS	-	(2)	-
IpAddress	-	-	-
Counter32	-	-	-
Counter64	-	-	-
Gauge32	(1)	-	-
TimeTicks	-	-	-

where:

(1) the range of permitted values may be refined by raising the lower-bounds, by reducing the upper-bounds, and/or by reducing the alternative value/range choices;

(2) the enumeration of named-values may be refined by removing one or more named-values (note that for BITS, a refinement may cause the enumerations to no longer be contiguous); or,

(3) the size in octets of the value may be refined by raising the lower-bounds, by reducing the upper-bounds, and/or by reducing the alternative size choices.

No other types of refinements can be specified in the SYNTAX clause. However, the DESCRIPTION clause is available to specify additional restrictions which can not be expressed in the SYNTAX clause. Further details on (and examples of) sub-typing are provided in Appendix A.

10. Extending an Information Module

As experience is gained with an information module, it may be desirable to revise that information module. However, changes are not allowed if they have any potential to cause interoperability problems "over the wire" between an implementation using an original specification and an implementation using an updated specification(s).

For any change, the invocation of the MODULE-IDENTITY macro must be updated to include information about the revision: specifically, updating the LAST-UPDATED clause, adding a pair of REVISION and DESCRIPTION clauses (see section 5.5), and making any necessary changes to existing clauses, including the ORGANIZATION and CONTACT-INFO clauses.

Note that any definition contained in an information module is available to be IMPORT-ed by any other information module, and is referenced in an IMPORTS clause via the module name. Thus, a module name should not be changed. Specifically, the module name (e.g., "FIZBIN-MIB" in the example of Section 5.7) should not be changed when revising an information module (except to correct typographical errors), and definitions should not be moved from one information module to another.

Also note that obsolete definitions must not be removed from MIB modules since their descriptors may still be referenced by other information modules, and the OBJECT IDENTIFIERs used to name them must never be re-assigned.

10.1. Object Assignments

If any non-editorial change is made to any clause of a object assignment, then the OBJECT IDENTIFIER value associated with that object assignment must also be changed, along with its associated descriptor.

10.2. Object Definitions

An object definition may be revised in any of the following ways:

(1) A SYNTAX clause containing an enumerated INTEGER may have new enumerations added or existing labels changed. Similarly, named bits may be added or existing labels changed for the BITS construct.

(2) The value of a SYNTAX clause may be replaced by a textual convention, providing the textual convention is defined to use the same primitive ASN.1 type, has the same set of values, and has identical semantics.

(3) A STATUS clause value of "current" may be revised as "deprecated" or "obsolete". Similarly, a STATUS clause value of "deprecated" may be revised as "obsolete". When making such a change, the DESCRIPTION clause should be updated to explain the rationale.

(4) A DEFVAL clause may be added or updated.

(5) A REFERENCE clause may be added or updated.

(6) A UNITS clause may be added.

(7) A conceptual row may be augmented by adding new columnar objects at the end of the row, and making the corresponding update to the SEQUENCE definition.

(8) Clarifications and additional information may be included in the DESCRIPTION clause.

(9) Entirely new objects may be defined, named with previously unassigned OBJECT IDENTIFIER values.

Otherwise, if the semantics of any previously defined object are changed (i.e., if a non-editorial change is made to any clause other than those specifically allowed above), then the OBJECT IDENTIFIER value associated with that object must also be changed.

McCloghrie, et al. Standards Track [Page 38]

RFC 2578 SMIv2 April 1999

Note that changing the descriptor associated with an existing object is considered a semantic change, as these strings may be used in an IMPORTS statement.

10.3. Notification Definitions

A notification definition may be revised in any of the following ways:

(1) A REFERENCE clause may be added or updated.

(2) A STATUS clause value of "current" may be revised as "deprecated" or "obsolete". Similarly, a STATUS clause value of "deprecated" may be revised as "obsolete". When making such a change, the

DESCRIPTION clause should be updated to explain the rationale.

(3) A DESCRIPTION clause may be clarified.

Otherwise, if the semantics of any previously defined notification are changed (i.e., if a non-editorial change is made to any clause other those specifically allowed above), then the OBJECT IDENTIFIER value associated with that notification must also be changed.

Note that changing the descriptor associated with an existing notification is considered a semantic change, as these strings may be used in an IMPORTS statement.

11. Appendix A: Detailed Sub-typing Rules

11.1. Syntax Rules

The syntax rules for sub-typing are given below. Note that while
this syntax is based on ASN.1, it includes some extensions beyond
what is allowed in ASN.1, and a number of ASN.1 constructs are not
allowed by this syntax.

```
<integerSubType>
   ::= <empty>
     | "(" <range> ["|" <range>]... ")"

<octetStringSubType>
   ::= <empty>
     | "(" "SIZE" "(" <range> ["|" <range>]... ")" ")"

<range>
   ::= <value>
     | <value> ".." <value>

<value>
   ::= "-" <number>
     | <number>
     | <hexString>
     | <binString>

where:
   <empty>     is the empty string
   <number>    is a non-negative integer
   <hexString> is a hexadecimal string (e.g., '0F0F'H)
   <binString> is a binary string (e.g, '1010'B)

   <range> is further restricted as follows:
      - any <value> used in a SIZE clause must be non-negative.
      - when a pair of values is specified, the first value
        must be less than the second value.
      - when multiple ranges are specified, the ranges may
        not overlap but may touch. For example, (1..4 | 4..9)
        is invalid, and (1..4 | 5..9) is valid.
      - the ranges must be a subset of the maximum range of the
        base type.
```

11.2. Examples

Some examples of legal sub-typing:

```
Integer32 (-20..100)
Integer32 (0..100 | 300..500)
Integer32 (300..500 | 0..100)
Integer32 (0 | 2 | 4 | 6 | 8 | 10)
OCTET STRING (SIZE(0..100))
OCTET STRING (SIZE(0..100 | 300..500))
OCTET STRING (SIZE(0 | 2 | 4 | 6 | 8 | 10))
SYNTAX   TimeInterval (0..100)
SYNTAX   DisplayString (SIZE(0..32))
```

(Note the last two examples above are not valid in a TEXTUAL CONVENTION, see [3].)

Some examples of illegal sub-typing:

```
Integer32 (150..100)         -- first greater than second
Integer32 (0..100 | 50..500) -- ranges overlap
Integer32 (0 | 2 | 0 )       -- value duplicated
Integer32 (MIN..-1 | 1..MAX) -- MIN and MAX not allowed
Integer32 (SIZE (0..34))     -- must not use SIZE
OCTET STRING (0..100)        -- must use SIZE
OCTET STRING (SIZE(-10..100)) -- negative SIZE
```

12. Security Considerations

This document defines a language with which to write and read descriptions of management information. The language itself has no security impact on the Internet.

13. Editors' Addresses

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive

San Jose, CA 95134-1706
USA
Phone: +1 408 526 5260
EMail: kzm@cisco.com

David Perkins
SNMPinfo
3763 Benton Street
Santa Clara, CA 95051
USA
Phone: +1 408 221-8702
EMail: dperkins@snmpinfo.com

Juergen Schoenwaelder
TU Braunschweig
Bueltenweg 74/75
38106 Braunschweig
Germany
Phone: +49 531 391-3283
EMail: schoenw@ibr.cs.tu-bs.de

14. References

[1] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, (December, 1987).

[2] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[3] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M.

and S. Waldbusser, "Textual Conventions for SMIv2", STD 58,
RFC 2579, April 1999.

[4] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8825, (December, 1987).

[5] The SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1907, January 1996.

[6] The SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

15. Full Copyright Statement

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

McCloghrie, et al.   Standards Track   [Page 43]

Found in path(s):

* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/ref/rfc/v2c/rfc2578.txt

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

```
RFC1158-MIB DEFINITIONS ::= BEGIN

IMPORTS
    mgmt, OBJECT-TYPE, NetworkAddress, IpAddress,
    Counter, Gauge, TimeTicks
        FROM RFC1155-SMI;

DisplayString ::=
        OCTET STRING
```

```
mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }    -- MIB-II
           -- (same prefix as MIB-I)

system     OBJECT IDENTIFIER ::= { mib-2 1 }
interfaces OBJECT IDENTIFIER ::= { mib-2 2 }
at         OBJECT IDENTIFIER ::= { mib-2 3 }
ip         OBJECT IDENTIFIER ::= { mib-2 4 }
icmp       OBJECT IDENTIFIER ::= { mib-2 5 }
tcp        OBJECT IDENTIFIER ::= { mib-2 6 }
udp        OBJECT IDENTIFIER ::= { mib-2 7 }
egp        OBJECT IDENTIFIER ::= { mib-2 8 }
-- cmot    OBJECT IDENTIFIER ::= { mib-2 9 }
transmission OBJECT IDENTIFIER ::= { mib-2 10 }
snmp       OBJECT IDENTIFIER ::= { mib-2 11 }


-- object types

-- the System group

sysDescr OBJECT-TYPE
       SYNTAX  DisplayString (SIZE (0..255))
       ACCESS  read-only
       STATUS  mandatory
       ::= { system 1 }

sysObjectID OBJECT-TYPE
       SYNTAX  OBJECT IDENTIFIER
       ACCESS  read-only
       STATUS  mandatory
       ::= { system 2 }

sysUpTime OBJECT-TYPE
       SYNTAX  TimeTicks
       ACCESS  read-only
       STATUS  mandatory
       ::= { system 3 }

sysContact OBJECT-TYPE
       SYNTAX  DisplayString (SIZE (0..255))
       ACCESS  read-write
       STATUS  mandatory
       ::= { system 4 }

sysName OBJECT-TYPE
       SYNTAX  DisplayString (SIZE (0..255))
       ACCESS  read-write
       STATUS  mandatory
```

```
      ::= { system 5 }

sysLocation OBJECT-TYPE
      SYNTAX  DisplayString (SIZE (0..255))
      ACCESS  read-only
      STATUS  mandatory
      ::= { system 6 }

sysServices OBJECT-TYPE
      SYNTAX  INTEGER (0..127)
      ACCESS  read-only
      STATUS  mandatory
      ::= { system 7 }


-- the Interfaces group

ifNumber OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      ::= { interfaces 1 }

-- the Interfaces table

ifTable OBJECT-TYPE
      SYNTAX  SEQUENCE OF IfEntry
      ACCESS  read-only
      STATUS  mandatory
      ::= { interfaces 2 }

ifEntry OBJECT-TYPE
      SYNTAX  IfEntry
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifTable 1 }

IfEntry ::= SEQUENCE {
   ifIndex
      INTEGER,
   ifDescr
      DisplayString,
   ifType
      INTEGER,
   ifMtu
      INTEGER,
   ifSpeed
      Gauge,
```

```
    ifPhysAddress
        OCTET STRING,
    ifAdminStatus
        INTEGER,
    ifOperStatus
        INTEGER,
    ifLastChange
        TimeTicks,
    ifInOctets
        Counter,
    ifInUcastPkts
        Counter,
    ifInNUcastPkts
        Counter,
    ifInDiscards
        Counter,
    ifInErrors
        Counter,
    ifInUnknownProtos
        Counter,
    ifOutOctets
        Counter,
    ifOutUcastPkts
        Counter,
    ifOutNUcastPkts
        Counter,
    ifOutDiscards
        Counter,
    ifOutErrors
        Counter,
    ifOutQLen
        Gauge,
    ifSpecific
        OBJECT IDENTIFIER
}

ifIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 1 }

ifDescr OBJECT-TYPE
        SYNTAX  DisplayString (SIZE (0..255))
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 2 }
```

```
ifType OBJECT-TYPE
    SYNTAX  INTEGER {
            other(1),          -- none of the
                               -- following
            regular1822(2),
            hdh1822(3),
            ddn-x25(4),
            rfc877-x25(5),
            ethernet-csmacd(6),
            iso88023-csmacd(7),
            iso88024-tokenBus(8),
            iso88025-tokenRing(9),
            iso88026-man(10),
            starLan(11),
            proteon-10Mbit(12),
            proteon-80Mbit(13),
            hyperchannel(14),
            fddi(15),
            lapb(16),
            sdlc(17),
            t1-carrier(18),
            cept(19),      -- european
                         --equivalent of T-1
            basicISDN(20),
            primaryISDN(21),
                               -- proprietary
                               -- serial
            propPointToPointSerial(22),
            terminalServer-asyncPort(23),
            softwareLoopback(24),
            eon(25),           -- CLNP over IP
            ethernet-3Mbit(26),
            nsip(27),          -- XNS over IP
            slip(28)           -- generic SLIP
      }
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 3 }

ifMtu OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 4 }

ifSpeed OBJECT-TYPE
    SYNTAX  Gauge
    ACCESS  read-only
```

```
    STATUS  mandatory
    ::= { ifEntry 5 }

ifPhysAddress OBJECT-TYPE
    SYNTAX  OCTET STRING
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 6 }

ifAdminStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                up(1),     -- ready to pass packets
                down(2),
                testing(3) -- in some test mode
            }
    ACCESS  read-write
    STATUS  mandatory
    ::= { ifEntry 7 }

ifOperStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                up(1),     -- ready to pass packets
                down(2),
                testing(3) -- in some test mode
            }
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 8 }

ifLastChange OBJECT-TYPE
    SYNTAX  TimeTicks
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 9 }

ifInOctets OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 10 }

ifInUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { ifEntry 11 }

ifInNUcastPkts OBJECT-TYPE
```

```
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 12 }

ifInDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 13 }

ifInErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 14 }

ifInUnknownProtos OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 15 }

ifOutOctets OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 16 }

ifOutUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 17 }

ifOutNUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 18 }

ifOutDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ifEntry 19 }

ifOutErrors OBJECT-TYPE
```

```
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 20 }

ifOutQLen OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 21 }

ifSpecific OBJECT-TYPE
        SYNTAX  OBJECT IDENTIFIER
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 22 }

nullSpecific OBJECT IDENTIFIER ::= { 0 0 }

-- the Address Translation group (deprecated)

atTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF AtEntry
        ACCESS  read-write
        STATUS  deprecated
        ::= { at 1 }

atEntry OBJECT-TYPE
        SYNTAX  AtEntry
        ACCESS  read-write
        STATUS  deprecated
        ::= { atTable 1 }

AtEntry ::= SEQUENCE {
    atIfIndex
        INTEGER,
    atPhysAddress
        OCTET STRING,
    atNetAddress
        NetworkAddress
}

atIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  deprecated
        ::= { atEntry 1 }
```

```
atPhysAddress OBJECT-TYPE
    SYNTAX  OCTET STRING
    ACCESS  read-write
    STATUS  deprecated
    ::= { atEntry 2 }

atNetAddress OBJECT-TYPE
    SYNTAX  NetworkAddress
    ACCESS  read-write
    STATUS  deprecated
    ::= { atEntry 3 }


-- the IP group

ipForwarding OBJECT-TYPE
    SYNTAX  INTEGER {
                gateway(1), -- entity forwards
                            -- datagrams
                host(2)     -- entity does NOT
                            -- forward datagrams
            }
    ACCESS  read-write
    STATUS  mandatory
    ::= { ip 1 }

ipDefaultTTL OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    ::= { ip 2 }

ipInReceives OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 3 }

ipInHdrErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 4 }

ipInAddrErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
```

```
    ::= { ip 5 }

ipForwDatagrams OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 6 }

ipInUnknownProtos OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 7 }

ipInDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 8 }

ipInDelivers OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 9 }

ipOutRequests OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 10 }

ipOutDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 11 }

ipOutNoRoutes OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 12 }

ipReasmTimeout OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
```

```
    ::= { ip 13 }

ipReasmReqds OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 14 }

ipReasmOKs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 15 }

ipReasmFails OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 16 }

ipFragOKs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 17 }

ipFragFails OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 18 }

ipFragCreates OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 19 }

-- the IP Interface table

ipAddrTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IpAddrEntry
    ACCESS  read-only
    STATUS  mandatory
    ::= { ip 20 }

ipAddrEntry OBJECT-TYPE
    SYNTAX  IpAddrEntry
```

```
      ACCESS  read-only
      STATUS  mandatory
      ::= { ipAddrTable 1 }

IpAddrEntry ::= SEQUENCE {
   ipAdEntAddr
      IpAddress,
   ipAdEntIfIndex
      INTEGER,
   ipAdEntNetMask
      IpAddress,
   ipAdEntBcastAddr
      INTEGER,
   ipAdEntReasmMaxSize
      INTEGER (0..65535)
}

ipAdEntAddr OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-only
      STATUS  mandatory
      ::=  { ipAddrEntry 1 }

ipAdEntIfIndex OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      ::=  { ipAddrEntry 2 }

ipAdEntNetMask OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-only
      STATUS  mandatory
      ::=  { ipAddrEntry 3 }

ipAdEntBcastAddr OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      ::= { ipAddrEntry 4 }

ipAdEntReasmMaxSize OBJECT-TYPE
      SYNTAX  INTEGER (0..65535)
      ACCESS  read-only
      STATUS  mandatory
      ::= { ipAddrEntry 5 }

-- the IP Routing table
```

```
ipRoutingTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IpRouteEntry
    ACCESS  read-write
    STATUS  mandatory
    ::= { ip 21 }

ipRouteEntry OBJECT-TYPE
    SYNTAX  IpRouteEntry
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRoutingTable 1 }

IpRouteEntry ::= SEQUENCE {
  ipRouteDest
    IpAddress,
  ipRouteIfIndex
    INTEGER,
  ipRouteMetric1
    INTEGER,
  ipRouteMetric2
    INTEGER,
  ipRouteMetric3
    INTEGER,
  ipRouteMetric4
    INTEGER,
  ipRouteNextHop
    IpAddress,
  ipRouteType
    INTEGER,
  ipRouteProto
    INTEGER,
  ipRouteAge
    INTEGER,
  ipRouteMask
    IpAddress
}

ipRouteDest OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRouteEntry 1 }

ipRouteIfIndex  OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
```

```
    ::= { ipRouteEntry 2 }

ipRouteMetric1 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRouteEntry 3 }

ipRouteMetric2 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRouteEntry 4 }

ipRouteMetric3 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRouteEntry 5 }

ipRouteMetric4 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRouteEntry 6 }

ipRouteNextHop OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRouteEntry 7 }

ipRouteType OBJECT-TYPE
    SYNTAX  INTEGER {
                other(1),    -- none of the following

                invalid(2),  -- an invalidated route

                             -- route to directly
                direct(3),   -- connected
                             -- (sub-)network

                             -- route to a non-local
                remote(4)    -- host/network/
                             -- sub-network
        }
    ACCESS  read-write
    STATUS  mandatory
```

```
    ::= { ipRouteEntry 8 }

ipRouteProto OBJECT-TYPE
    SYNTAX  INTEGER {
            other(1),    -- none of the following

                         -- non-protocol
                         -- information

                         --   e.g., manually
            local(2),    --   configured entries

                         -- set via a network
            netmgmt(3),  --   management protocol

                         -- obtained via ICMP,
            icmp(4),     --   e.g., Redirect

                         -- the following are
                         -- gateway routing
                         -- protocols
            egp(5),
            ggp(6),
            hello(7),
            rip(8),
            is-is(9),
            es-is(10),
            ciscoIgrp(11),
            bbnSpfIgp(12),
            ospf(13),
            bgp(14)
        }
    ACCESS  read-only
    STATUS  mandatory
    ::= { ipRouteEntry 9 }

ipRouteAge OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRouteEntry 10 }

ipRouteMask OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS  mandatory
    ::= { ipRouteEntry 11 }
```

```
-- the IP Address Translation tables

ipNetToMediaTable OBJECT-TYPE
      SYNTAX  SEQUENCE OF IpNetToMediaEntry
      ACCESS  read-write
      STATUS  mandatory
      ::= { ip 22 }

ipNetToMediaEntry OBJECT-TYPE
      SYNTAX  IpNetToMediaEntry
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipNetToMediaTable 1 }

IpNetToMediaEntry ::= SEQUENCE {
   ipNetToMediaIfIndex
      INTEGER,
   ipNetToMediaPhysAddress
      OCTET STRING,
   ipNetToMediaNetAddress
      IpAddress,
   ipNetToMediaType
      INTEGER
}

ipNetToMediaIfIndex OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipNetToMediaEntry 1 }

ipNetToMediaPhysAddress OBJECT-TYPE
      SYNTAX  OCTET STRING
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipNetToMediaEntry 2 }

ipNetToMediaNetAddress OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipNetToMediaEntry 3 }

ipNetToMediaType OBJECT-TYPE
      SYNTAX  INTEGER {
                 other(1),   -- none of the following

                 invalid(2), -- an invalidated mapping
```

```
                dynamic(3), -- connected (sub-)network

                static(4)
            }
        ACCESS  read-write
        STATUS  mandatory
        ::= { ipNetToMediaEntry 4 }

-- the ICMP group

icmpInMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 1 }

icmpInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 2 }

icmpInDestUnreachs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 3 }

icmpInTimeExcds OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 4 }

icmpInParmProbs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 5 }

icmpInSrcQuenchs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 6 }

icmpInRedirects OBJECT-TYPE
        SYNTAX  Counter
```

```
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 7 }

icmpInEchos OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 8 }

icmpInEchoReps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 9 }

icmpInTimestamps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 10 }

icmpInTimestampReps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 11 }

icmpInAddrMasks OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 12 }

icmpInAddrMaskReps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 13 }

icmpOutMsgs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 14 }

icmpOutErrors OBJECT-TYPE
    SYNTAX  Counter
```

```
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 15 }

icmpOutDestUnreachs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 16 }

icmpOutTimeExcds OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 17 }

icmpOutParmProbs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 18 }

icmpOutSrcQuenchs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 19 }

icmpOutRedirects OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 20 }

icmpOutEchos OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 21 }

icmpOutEchoReps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { icmp 22 }

icmpOutTimestamps OBJECT-TYPE
    SYNTAX  Counter
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 23 }

    icmpOutTimestampReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 24 }

    icmpOutAddrMasks OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 25 }

    icmpOutAddrMaskReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 26 }


    -- the TCP group

    tcpRtoAlgorithm OBJECT-TYPE
        SYNTAX  INTEGER {
                    other(1),    -- none of the following
                    constant(2), -- a constant rto
                    rsre(3),     -- MIL-STD-1778,
                                 -- Appendix B
                    vanj(4)      -- Van Jacobson's
                                 -- algorithm
                }
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 1 }

    tcpRtoMin OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 2 }

    tcpRtoMax OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
```

```
      ::= { tcp 3 }

tcpMaxConn OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      ::= { tcp 4 }

tcpActiveOpens OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { tcp 5 }

tcpPassiveOpens OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { tcp 6 }

tcpAttemptFails OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { tcp 7 }

tcpEstabResets OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { tcp 8 }

tcpCurrEstab OBJECT-TYPE
      SYNTAX  Gauge
      ACCESS  read-only
      STATUS  mandatory
      ::= { tcp 9 }

tcpInSegs OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { tcp 10 }

tcpOutSegs OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
```

```
    ::= { tcp 11 }

tcpRetransSegs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcp 12 }

-- the TCP connections table

tcpConnTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF TcpConnEntry
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
    SYNTAX  TcpConnEntry
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcpConnTable 1 }

TcpConnEntry ::= SEQUENCE {
  tcpConnState
    INTEGER,
  tcpConnLocalAddress
    IpAddress,
  tcpConnLocalPort
    INTEGER (0..65535),
  tcpConnRemAddress
    IpAddress,
  tcpConnRemPort
    INTEGER (0..65535)
}

tcpConnState OBJECT-TYPE
    SYNTAX  INTEGER {
            closed(1),
            listen(2),
            synSent(3),
            synReceived(4),
            established(5),
            finWait1(6),
            finWait2(7),
            closeWait(8),
            lastAck(9),
            closing(10),
            timeWait(11)
```

```
        }
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcpConnEntry 1 }

tcpConnLocalAddress OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcpConnEntry 2 }

tcpConnLocalPort OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcpConnEntry 5 }

-- additional TCP variables

tcpInErrs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcp 14 }

tcpOutRsts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcp 15 }


-- the UDP group

udpInDatagrams OBJECT-TYPE
    SYNTAX  Counter
```

```
      ACCESS  read-only
      STATUS  mandatory
      ::= { udp 1 }

udpNoPorts OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { udp 2 }

udpInErrors OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { udp 3 }

udpOutDatagrams OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { udp 4 }

-- the UDP listener table

udpTable OBJECT-TYPE
      SYNTAX  SEQUENCE OF UdpEntry
      ACCESS  read-only
      STATUS  mandatory
      ::= { udp 5 }

udpEntry OBJECT-TYPE
      SYNTAX  UdpEntry
      ACCESS  read-only
      STATUS  mandatory
      ::= { udpTable 1 }

UdpEntry ::= SEQUENCE {
   udpLocalAddress
      IpAddress,
   udpLocalPort
      INTEGER (0..65535)
}

udpLocalAddress OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-only
      STATUS  mandatory
      ::= { udpEntry 1 }
```

```
udpLocalPort OBJECT-TYPE
    SYNTAX  INTEGER (0..65535)
    ACCESS  read-only
    STATUS  mandatory
    ::= { udpEntry 2 }

-- the EGP group

egpInMsgs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egp 1 }

egpInErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egp 2 }

egpOutMsgs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egp 3 }

egpOutErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egp 4 }

-- the EGP Neighbor table

egpNeighTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF EgpNeighEntry
    ACCESS  read-only
    STATUS  mandatory
    ::= { egp 5 }

egpNeighEntry OBJECT-TYPE
    SYNTAX  EgpNeighEntry
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighTable 1 }

EgpNeighEntry ::= SEQUENCE {
```

```
  egpNeighState
     INTEGER,
  egpNeighAddr
     IpAddress,
  egpNeighAs
     INTEGER,
  egpNeighInMsgs
     Counter,
  egpNeighInErrs
     Counter,
  egpNeighOutMsgs
     Counter,
  egpNeighOutErrs
     Counter,
  egpNeighInErrMsgs
     Counter,
  egpNeighOutErrMsgs
     Counter,
  egpNeighStateUps
     Counter,
  egpNeighStateDowns
     Counter,
  egpNeighIntervalHello
     INTEGER,
  egpNeighIntervalPoll
     INTEGER,
  egpNeighMode
     INTEGER,
  egpNeighEventTrigger
     INTEGER
}

egpNeighState OBJECT-TYPE
     SYNTAX  INTEGER {
             idle(1),
             acquisition(2),
             down(3),
             up(4),
             cease(5)
           }
     ACCESS  read-only
     STATUS  mandatory
     ::= { egpNeighEntry 1 }

egpNeighAddr OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
```

```
    ::= { egpNeighEntry 2 }

egpNeighAs OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 3 }

egpNeighInMsgs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 4 }

egpNeighInErrs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 5 }

egpNeighOutMsgs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 6 }

egpNeighOutErrs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 7 }

egpNeighInErrMsgs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 8 }

egpNeighOutErrMsgs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 9 }

egpNeighStateUps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
```

```
    ::= { egpNeighEntry 10 }

egpNeighStateDowns OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 11 }

egpNeighIntervalHello OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 12 }

egpNeighIntervalPoll OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 13 }

egpNeighMode OBJECT-TYPE
    SYNTAX  INTEGER {
              active(1),
              passive(2)
           }
    ACCESS  read-only
    STATUS  mandatory
    ::= { egpNeighEntry 14 }

egpNeighEventTrigger OBJECT-TYPE
    SYNTAX  INTEGER {
              start(1),
              stop(2)
           }
    ACCESS  read-write
    STATUS  mandatory
    ::= { egpNeighEntry 15 }

-- additional EGP variables

egpAs OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    ::= { egp 6 }


-- the Transmission group (empty at present)
```

```
-- the SNMP group

snmpInPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 1 }

snmpOutPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 2 }

snmpInBadVersions OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 3 }

snmpInBadCommunityNames OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 4 }

snmpInBadCommunityUses OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 5 }

snmpInASNParseErrs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 6 }

snmpInBadTypes OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 7 }

snmpInTooBigs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
```

```
    STATUS  mandatory
    ::=  { snmp 8 }

snmpInNoSuchNames OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 9 }

snmpInBadValues OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 10 }

snmpInReadOnlys OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 11 }

snmpInGenErrs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 12 }

snmpInTotalReqVars OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 13 }

snmpInTotalSetVars OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 14 }

snmpInGetRequests OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 15 }

snmpInGetNexts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
```

```
    STATUS  mandatory
    ::=  { snmp 16 }

snmpInSetRequests OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 17 }

snmpInGetResponses OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 18 }

snmpInTraps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 19 }

snmpOutTooBigs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 20 }

snmpOutNoSuchNames OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 21 }

snmpOutBadValues OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 22 }

snmpOutReadOnlys OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 23 }

snmpOutGenErrs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
```

```
    STATUS  mandatory
    ::=  { snmp 24 }

snmpOutGetRequests OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 25 }

snmpOutGetNexts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 26 }

snmpOutSetRequests OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 27 }

snmpOutGetResponses OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 28 }

snmpOutTraps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::=  { snmp 29 }

snmpEnableAuthTraps OBJECT-TYPE
    SYNTAX  INTEGER {
                enabled(1),
                disabled(2)
            }
    ACCESS  read-write
    STATUS  mandatory
    ::=  { snmp 30 }

END
```

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/lib/mibs/RFC1158-MIB.mib
No license file was found, but licenses were detected in source scan.

Network Working Group U. Blumenthal
Request for Comments: 3414 B. Wijnen
STD: 62 Lucent Technologies
Obsoletes: 2574 December 2002
Category: Standards Track

User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document describes the User-based Security Model (USM) for Simple Network Management Protocol (SNMP) version 3 for use in the SNMP architecture. It defines the Elements of Procedure for providing SNMP message level security. This document also includes a Management Information Base (MIB) for remotely monitoring/managing the configuration parameters for this Security Model. This document obsoletes RFC 2574.

Table of Contents

1. Introduction

The Architecture for describing Internet Management Frameworks [RFC3411] describes that an SNMP engine is composed of:

1) a Dispatcher,
2) a Message Processing Subsystem,
3) a Security Subsystem, and
4) an Access Control Subsystem.

Applications make use of the services of these subsystems.

It is important to understand the SNMP architecture and the terminology of the architecture to understand where the Security Model described in this document fits into the architecture and interacts with other subsystems within the architecture. The reader is expected to have read and understood the description of the SNMP architecture, as defined in [RFC3411].

This memo describes the User-based Security Model as it is used within the SNMP Architecture. The main idea is that we use the traditional concept of a user (identified by a userName) with which to associate security information.

This memo describes the use of HMAC-MD5-96 and HMAC-SHA-96 as the authentication protocols and the use of CBC-DES as the privacy protocol. The User-based Security Model however allows for other such protocols to be used instead of or concurrent with these protocols. Therefore, the description of HMAC-MD5-96, HMAC-SHA-96 and CBC-DES are in separate sections to reflect their self-contained nature and to indicate that they can be replaced or supplemented in the future.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.1. Threats

Several of the classical threats to network protocols are applicable to the network management problem and therefore would be applicable to any SNMP Security Model. Other threats are not applicable to the network management problem. This section discusses principal threats, secondary threats, and threats which are of lesser importance.

The principal threats against which this SNMP Security Model should provide protection are:

- Modification of Information The modification threat is the danger that some unauthorized entity may alter in-transit SNMP messages generated on behalf of an authorized principal in such a way as to effect unauthorized management operations, including falsifying the value of an object.

- Masquerade The masquerade threat is the danger that management operations not authorized for some user may be attempted by assuming the identity of another user that has the appropriate authorizations.

Two secondary threats are also identified. The Security Model defined in this memo provides limited protection against:

- Disclosure The disclosure threat is the danger of eavesdropping on the exchanges between managed agents and a management station.

Protecting against this threat may be required as a matter of local policy.

- Message Stream Modification The SNMP protocol is typically based upon a connection-less transport service which may operate over any sub-network service. The re-ordering, delay or replay of messages can and does occur through the natural operation of many such sub-network services. The message stream modification threat is the danger that messages may be maliciously re-ordered, delayed or replayed to an extent which is greater than can occur through the natural operation of a sub-network service, in order to effect unauthorized management operations.

There are at least two threats that an SNMP Security Model need not protect against. The security protocols defined in this memo do not provide protection against:

- Denial of Service This SNMP Security Model does not attempt to address the broad range of attacks by which service on behalf of authorized users is denied. Indeed, such denial-of-service attacks are in many cases indistinguishable from the type of network failures with which any viable network management protocol must cope as a matter of course.

- Traffic Analysis This SNMP Security Model does not attempt to address traffic analysis attacks. Indeed, many traffic patterns are predictable - devices may be managed on a regular basis by a relatively small number of management applications - and therefore there is no significant advantage afforded by protecting against traffic analysis.

1.2. Goals and Constraints

Based on the foregoing account of threats in the SNMP network management environment, the goals of this SNMP Security Model are as follows.

1) Provide for verification that each received SNMP message has not been modified during its transmission through the network.

2) Provide for verification of the identity of the user on whose behalf a received SNMP message claims to have been generated.

3) Provide for detection of received SNMP messages, which request or contain management information, whose time of generation was not recent.

4) Provide, when necessary, that the contents of each received SNMP message are protected from disclosure.

In addition to the principal goal of supporting secure network management, the design of this SNMP Security Model is also influenced by the following constraints:

1) When the requirements of effective management in times of network stress are inconsistent with those of security, the design of USM has given preference to the former.

2) Neither the security protocol nor its underlying security mechanisms should depend upon the ready availability of other network services (e.g., Network Time Protocol (NTP) or key management protocols).

3) A security mechanism should entail no changes to the basic SNMP network management philosophy.

1.3. Security Services

The security services necessary to support the goals of this SNMP Security Model are as follows:

- Data Integrity is the provision of the property that data has not been altered or destroyed in an unauthorized manner, nor have data sequences been altered to an extent greater than can occur non-maliciously.

- Data Origin Authentication is the provision of the property that the claimed identity of the user on whose behalf received data was originated is corroborated.

- Data Confidentiality is the provision of the property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

- Message timeliness and limited replay protection is the provision of the property that a message whose generation time is outside of a specified time window is not accepted. Note that message reordering is not dealt with and can occur in normal conditions too.

For the protocols specified in this memo, it is not possible to assure the specific originator of a received SNMP message; rather, it is the user on whose behalf the message was originated that is authenticated.

For these protocols, it not possible to obtain data integrity without data origin authentication, nor is it possible to obtain data origin authentication without data integrity. Further, there is no provision for data confidentiality without both data integrity and data origin authentication.

The security protocols used in this memo are considered acceptably secure at the time of writing. However, the procedures allow for new authentication and privacy methods to be specified at a future time if the need arises.

1.4. Module Organization

The security protocols defined in this memo are split in three different modules and each has its specific responsibilities such that together they realize the goals and security services described above:

- The authentication module MUST provide for:

  - Data Integrity,

  - Data Origin Authentication,

- The timeliness module MUST provide for:

  - Protection against message delay or replay (to an extent greater than can occur through normal operation).

- The privacy module MUST provide for

  - Protection against disclosure of the message payload.

The timeliness module is fixed for the User-based Security Model while there is provision for multiple authentication and/or privacy modules, each of which implements a specific authentication or privacy protocol respectively.

1.4.1. Timeliness Module

Section 3 (Elements of Procedure) uses the timeliness values in an SNMP message to do timeliness checking. The timeliness check is only performed if authentication is applied to the message. Since the complete message is checked for integrity, we can assume that the timeliness values in a message that passes the authentication module are trustworthy.

1.4.2. Authentication Protocol

Section 6 describes the HMAC-MD5-96 authentication protocol which is the first authentication protocol that MUST be supported with the User-based Security Model. Section 7 describes the HMAC-SHA-96 authentication protocol which is another authentication protocol that SHOULD be supported with the User-based Security Model. In the future additional or replacement authentication protocols may be defined as new needs arise.

The User-based Security Model prescribes that, if authentication is used, then the complete message is checked for integrity in the authentication module.

For a message to be authenticated, it needs to pass authentication check by the authentication module and the timeliness check which is a fixed part of this User-based Security model.

1.4.3. Privacy Protocol

Section 8 describes the CBC-DES Symmetric Encryption Protocol which is the first privacy protocol to be used with the User-based Security Model. In the future additional or replacement privacy protocols may be defined as new needs arise.

The User-based Security Model prescribes that the scopedPDU is protected from disclosure when a message is sent with privacy.

The User-based Security Model also prescribes that a message needs to be authenticated if privacy is in use.

1.5. Protection against Message Replay, Delay and Redirection

1.5.1. Authoritative SNMP Engine

In order to protect against message replay, delay and redirection, one of the SNMP engines involved in each communication is designated to be the authoritative SNMP engine. When an SNMP message contains a payload which expects a response (those messages that contain a Confirmed Class PDU [RFC3411]), then the receiver of such messages is authoritative. When an SNMP message contains a payload which does not expect a response (those messages that contain an Unconfirmed Class PDU [RFC3411]), then the sender of such a message is authoritative.

1.5.2. Mechanisms

The following mechanisms are used:

1) To protect against the threat of message delay or replay (to an extent greater than can occur through normal operation), a set of timeliness indicators (for the authoritative SNMP engine) are included in each message generated. An SNMP engine evaluates the timeliness indicators to determine if a received message is recent. An SNMP engine may evaluate the timeliness indicators to ensure that a received message is at least as recent as the last message it received from the same source. A non-authoritative SNMP engine uses received authentic messages to advance its notion of the timeliness indicators at the remote authoritative source.

   An SNMP engine MUST also use a mechanism to match incoming Responses to outstanding Requests and it MUST drop any Responses that do not match an outstanding request. For example, a msgID can be inserted in every message to cater for this functionality.

These mechanisms provide for the detection of authenticated messages whose time of generation was not recent.

This protection against the threat of message delay or replay does not imply nor provide any protection against unauthorized deletion or suppression of messages. Also, an SNMP engine may not be able to detect message reordering if all the messages involved are sent within the Time Window interval. Other mechanisms defined independently of the security protocol can also be used to detect the re-ordering replay, deletion, or suppression of messages containing Set operations (e.g., the MIB variable snmpSetSerialNo [RFC3418]).

2) Verification that a message sent to/from one authoritative SNMP engine cannot be replayed to/as-if-from another authoritative SNMP engine.

Included in each message is an identifier unique to the authoritative SNMP engine associated with the sender or intended recipient of the message.

A message containing an Unconfirmed Class PDU sent by an authoritative SNMP engine to one non-authoritative SNMP engine can potentially be replayed to another non-authoritative SNMP engine. The latter non-authoritative SNMP engine might (if it knows about the same userName with the same secrets at the authoritative SNMP engine) as a result update its notion of timeliness indicators of the authoritative SNMP engine, but that is not considered a threat. In this case, A Report or Response message will be discarded by the Message Processing Model, because there should not be an outstanding Request message. A Trap will possibly be accepted. Again, that is not considered a threat, because the communication was authenticated and timely. It is as if the authoritative SNMP engine was configured to start sending Traps to the second SNMP engine, which theoretically can happen without the knowledge of the second SNMP engine anyway. Anyway, the second SNMP engine may not expect to receive this Trap, but is allowed to see the management information contained in it.

3) Detection of messages which were not recently generated.

A set of time indicators are included in the message, indicating the time of generation. Messages without recent time indicators are not considered authentic. In addition, an SNMP engine MUST drop any Responses that do not match an outstanding request. This however is the responsibility of the Message Processing Model.

This memo allows the same user to be defined on multiple SNMP engines. Each SNMP engine maintains a value, snmpEngineID, which uniquely identifies the SNMP engine. This value is included in each message sent to/from the SNMP engine that is authoritative (see section 1.5.1). On receipt of a message, an authoritative SNMP engine checks the value to ensure that it is the intended recipient, and a non-authoritative SNMP engine uses the value to ensure that the message is processed using the correct state information.

Each SNMP engine maintains two values, snmpEngineBoots and snmpEngineTime, which taken together provide an indication of time at that SNMP engine. Both of these values are included in an authenticated message sent to/received from that SNMP engine. On receipt, the values are checked to ensure that the indicated

timeliness value is within a Time Window of the current time. The Time Window represents an administrative upper bound on acceptable delivery delay for protocol messages.

For an SNMP engine to generate a message which an authoritative SNMP engine will accept as authentic, and to verify that a message received from that authoritative SNMP engine is authentic, such an SNMP engine must first achieve timeliness synchronization with the authoritative SNMP engine. See section 2.3.

1.6. Abstract Service Interfaces

Abstract service interfaces have been defined to describe the conceptual interfaces between the various subsystems within an SNMP entity. Similarly a set of abstract service interfaces have been defined within the User-based Security Model (USM) to describe the conceptual interfaces between the generic USM services and the self-contained authentication and privacy services.

These abstract service interfaces are defined by a set of primitives that define the services provided and the abstract data elements that must be passed when the services are invoked. This section lists the primitives that have been defined for the User-based Security Model.

1.6.1. User-based Security Model Primitives for Authentication

The User-based Security Model provides the following internal primitives to pass data back and forth between the Security Model itself and the authentication service:

```
statusInformation =
  authenticateOutgoingMsg(
  IN   authKey                -- secret key for authentication
  IN   wholeMsg               -- unauthenticated complete message
  OUT  authenticatedWholeMsg  -- complete authenticated message
     )

statusInformation =
  authenticateIncomingMsg(
  IN   authKey                -- secret key for authentication
  IN   authParameters         -- as received on the wire
  IN   wholeMsg               -- as received on the wire
  OUT  authenticatedWholeMsg  -- complete authenticated message
     )
```


1.6.2. User-based Security Model Primitives for Privacy

The User-based Security Model provides the following internal primitives to pass data back and forth between the Security Model itself and the privacy service:

```
statusInformation =
  encryptData(
```

```
  IN    encryptKey             -- secret key for encryption
  IN    dataToEncrypt          -- data to encrypt (scopedPDU)
  OUT   encryptedData          -- encrypted data (encryptedPDU)
  OUT   privParameters         -- filled in by service provider
        )

 statusInformation =
  decryptData(
  IN    decryptKey             -- secret key for decrypting
  IN    privParameters         -- as received on the wire
  IN    encryptedData          -- encrypted data (encryptedPDU)
  OUT   decryptedData          -- decrypted data (scopedPDU)
        )
```

2. Elements of the Model

This section contains definitions required to realize the security model defined by this memo.

2.1. User-based Security Model Users

Management operations using this Security Model make use of a defined set of user identities. For any user on whose behalf management operations are authorized at a particular SNMP engine, that SNMP engine must have knowledge of that user. An SNMP engine that wishes to communicate with another SNMP engine must also have knowledge of a user known to that engine, including knowledge of the applicable attributes of that user.

A user and its attributes are defined as follows:

userName
   A string representing the name of the user.

securityName
   A human-readable string representing the user in a format that is Security Model independent. There is a one-to-one relationship between userName and securityName.

authProtocol

An indication of whether messages sent on behalf of this user can be authenticated, and if so, the type of authentication protocol which is used. Two such protocols are defined in this memo:

- the HMAC-MD5-96 authentication protocol.
- the HMAC-SHA-96 authentication protocol.

authKey

If messages sent on behalf of this user can be authenticated, the (private) authentication key for use with the authentication protocol. Note that a user's authentication key will normally be different at different authoritative SNMP engines. The authKey is not accessible via SNMP. The length requirements of the authKey are defined by the authProtocol in use.

authKeyChange and authOwnKeyChange

The only way to remotely update the authentication key. Does that in a secure manner, so that the update can be completed without the need to employ privacy protection.

privProtocol

An indication of whether messages sent on behalf of this user can be protected from disclosure, and if so, the type of privacy protocol which is used. One such protocol is defined in this memo: the CBC-DES Symmetric Encryption Protocol.

privKey

If messages sent on behalf of this user can be en/decrypted, the (private) privacy key for use with the privacy protocol. Note that a user's privacy key will normally be different at different authoritative SNMP engines. The privKey is not accessible via SNMP. The length requirements of the privKey are defined by the privProtocol in use.

privKeyChange and privOwnKeyChange

The only way to remotely update the encryption key. Does that in a secure manner, so that the update can be completed without the need to employ privacy protection.

2.2. Replay Protection

Each SNMP engine maintains three objects:

- snmpEngineID, which (at least within an administrative domain) uniquely and unambiguously identifies an SNMP engine.

- snmpEngineBoots, which is a count of the number of times the SNMP engine has re-booted/re-initialized since snmpEngineID was last configured; and,

- snmpEngineTime, which is the number of seconds since the snmpEngineBoots counter was last incremented.

Each SNMP engine is always authoritative with respect to these objects in its own SNMP entity. It is the responsibility of a non-authoritative SNMP engine to synchronize with the authoritative SNMP engine, as appropriate.

An authoritative SNMP engine is required to maintain the values of its snmpEngineID and snmpEngineBoots in non-volatile storage.

2.2.1. msgAuthoritativeEngineID

The msgAuthoritativeEngineID value contained in an authenticated message is used to defeat attacks in which messages from one SNMP engine to another SNMP engine are replayed to a different SNMP engine. It represents the snmpEngineID at the authoritative SNMP engine involved in the exchange of the message.

When an authoritative SNMP engine is first installed, it sets its local value of snmpEngineID according to a enterprise-specific algorithm (see the definition of the Textual Convention for SnmpEngineID in the SNMP Architecture document [RFC3411]).

2.2.2. msgAuthoritativeEngineBoots and msgAuthoritativeEngineTime

The msgAuthoritativeEngineBoots and msgAuthoritativeEngineTime values contained in an authenticated message are used to defeat attacks in which messages are replayed when they are no longer valid. They represent the snmpEngineBoots and snmpEngineTime values at the authoritative SNMP engine involved in the exchange of the message.

Through use of snmpEngineBoots and snmpEngineTime, there is no requirement for an SNMP engine to have a non-volatile clock which ticks (i.e., increases with the passage of time) even when the SNMP engine is powered off. Rather, each time an SNMP engine re-boots, it retrieves, increments, and then stores snmpEngineBoots in non-volatile storage, and resets snmpEngineTime to zero.

When an SNMP engine is first installed, it sets its local values of snmpEngineBoots and snmpEngineTime to zero. If snmpEngineTime ever reaches its maximum value (2147483647), then snmpEngineBoots is incremented as if the SNMP engine has re-booted and snmpEngineTime is reset to zero and starts incrementing again.

Each time an authoritative SNMP engine re-boots, any SNMP engines holding that authoritative SNMP engine's values of snmpEngineBoots and snmpEngineTime need to re-synchronize prior to sending correctly authenticated messages to that authoritative SNMP engine (see Section 2.3 for (re-)synchronization procedures). Note, however, that the procedures do provide for a notification to be accepted as authentic by a receiving SNMP engine, when sent by an authoritative SNMP engine which has re-booted since the receiving SNMP engine last (re-)synchronized.

If an authoritative SNMP engine is ever unable to determine its latest snmpEngineBoots value, then it must set its snmpEngineBoots value to 2147483647.

Whenever the local value of snmpEngineBoots has the value 2147483647 it latches at that value and an authenticated message always causes an notInTimeWindow authentication failure.

In order to reset an SNMP engine whose snmpEngineBoots value has reached the value 2147483647, manual intervention is required. The engine must be physically visited and re-configured, either with a new snmpEngineID value, or with new secret values for the authentication and privacy protocols of all users known to that SNMP engine. Note that even if an SNMP engine re-boots once a second that it would still take approximately 68 years before the max value of 2147483647 would be reached.

2.2.3. Time Window

The Time Window is a value that specifies the window of time in which a message generated on behalf of any user is valid. This memo specifies that the same value of the Time Window, 150 seconds, is used for all users.

2.3. Time Synchronization

Time synchronization, required by a non-authoritative SNMP engine in order to proceed with authentic communications, has occurred when the non-authoritative SNMP engine has obtained a local notion of the authoritative SNMP engine's values of snmpEngineBoots and snmpEngineTime from the authoritative SNMP engine. These values must be (and remain) within the authoritative SNMP engine's Time Window. So the local notion of the authoritative SNMP engine's values must be kept loosely synchronized with the values stored at the authoritative SNMP engine. In addition to keeping a local copy of snmpEngineBoots and snmpEngineTime from the authoritative SNMP engine, a non-authoritative SNMP engine must also keep one

local variable, latestReceivedEngineTime. This value records the highest value of snmpEngineTime that was received by the non-authoritative SNMP engine from the authoritative SNMP engine and is used to eliminate the possibility of replaying messages that would prevent the non-authoritative SNMP engine's notion of the snmpEngineTime from advancing.

A non-authoritative SNMP engine must keep local notions of these values (snmpEngineBoots, snmpEngineTime and latestReceivedEngineTime) for each authoritative SNMP engine with which it wishes to communicate. Since each authoritative SNMP engine is uniquely and unambiguously identified by its value of snmpEngineID, the non-authoritative SNMP engine may use this value as a key in order to cache its local notions of these values.

Time synchronization occurs as part of the procedures of receiving an SNMP message (Section 3.2, step 7b). As such, no explicit time synchronization procedure is required by a non-authoritative SNMP engine. Note, that whenever the local value of snmpEngineID is changed (e.g., through discovery) or when secure communications are first established with an authoritative SNMP engine, the local values of snmpEngineBoots and latestReceivedEngineTime should be set to zero. This will cause the time synchronization to occur when the next authentic message is received.

2.4. SNMP Messages Using this Security Model

The syntax of an SNMP message using this Security Model adheres to the message format defined in the version-specific Message Processing Model document (for example [RFC3412]).

The field msgSecurityParameters in SNMPv3 messages has a data type of OCTET STRING. Its value is the BER serialization of the following ASN.1 sequence:

```
USMSecurityParametersSyntax DEFINITIONS IMPLICIT TAGS ::= BEGIN

   UsmSecurityParameters ::=
       SEQUENCE {
        -- global User-based security parameters
          msgAuthoritativeEngineID     OCTET STRING,
          msgAuthoritativeEngineBoots  INTEGER (0..2147483647),
          msgAuthoritativeEngineTime   INTEGER (0..2147483647),
          msgUserName                  OCTET STRING (SIZE(0..32)),
        -- authentication protocol specific parameters
          msgAuthenticationParameters  OCTET STRING,
        -- privacy protocol specific parameters
          msgPrivacyParameters         OCTET STRING
```



```
       }
END
```

The fields of this sequence are:

- The msgAuthoritativeEngineID specifies the snmpEngineID of the authoritative SNMP engine involved in the exchange of the message.

- The msgAuthoritativeEngineBoots specifies the snmpEngineBoots value at the authoritative SNMP engine involved in the exchange of the message.

- The msgAuthoritativeEngineTime specifies the snmpEngineTime value at the authoritative SNMP engine involved in the exchange of the message.

- The msgUserName specifies the user (principal) on whose behalf the message is being exchanged. Note that a zero-length userName will not match any user, but it can be used for snmpEngineID discovery.

- The msgAuthenticationParameters are defined by the authentication protocol in use for the message, as defined by the usmUserAuthProtocol column in the user's entry in the usmUserTable.

- The msgPrivacyParameters are defined by the privacy protocol in use for the message, as defined by the usmUserPrivProtocol column in the user's entry in the usmUserTable).

See appendix A.4 for an example of the BER encoding of field msgSecurityParameters.

2.5. Services provided by the User-based Security Model

This section describes the services provided by the User-based Security Model with their inputs and outputs.

The services are described as primitives of an abstract service interface and the inputs and outputs are described as abstract data elements as they are passed in these abstract service primitives.

2.5.1. Services for Generating an Outgoing SNMP Message

When the Message Processing (MP) Subsystem invokes the User-based Security module to secure an outgoing SNMP message, it must use the appropriate service as provided by the Security module. These two services are provided:

1) A service to generate a Request message. The abstract service primitive is:

```
statusInformation =            -- success or errorIndication
  generateRequestMsg(
  IN   messageProcessingModel  -- typically, SNMP version
  IN   globalData              -- message header, admin data
  IN   maxMessageSize          -- of the sending SNMP entity
```

```
    IN   securityModel          -- for the outgoing message
    IN   securityEngineID       -- authoritative SNMP entity
    IN   securityName           -- on behalf of this principal
    IN   securityLevel          -- Level of Security requested
    IN   scopedPDU              -- message (plaintext) payload
    OUT  securityParameters     -- filled in by Security Module
    OUT  wholeMsg               -- complete generated message
    OUT  wholeMsgLength         -- length of generated message
         )
```

2) A service to generate a Response message. The abstract service primitive is:

```
   statusInformation =          -- success or errorIndication
    generateResponseMsg(
    IN   messageProcessingModel  -- typically, SNMP version
    IN   globalData             -- message header, admin data
    IN   maxMessageSize         -- of the sending SNMP entity
    IN   securityModel          -- for the outgoing message
    IN   securityEngineID       -- authoritative SNMP entity
    IN   securityName           -- on behalf of this principal
    IN   securityLevel          -- Level of Security requested
    IN   scopedPDU              -- message (plaintext) payload
    IN   securityStateReference  -- reference to security state
                                -- information from original
                                -- request
    OUT  securityParameters     -- filled in by Security Module
    OUT  wholeMsg               -- complete generated message
    OUT  wholeMsgLength         -- length of generated message
         )
```

The abstract data elements passed as parameters in the abstract service primitives are as follows:

statusInformation
   An indication of whether the encoding and securing of the message was successful. If not it is an indication of the problem.

messageProcessingModel
The SNMP version number for the message to be generated. This data is not used by the User-based Security module.

globalData
The message header (i.e., its administrative information). This data is not used by the User-based Security module.

maxMessageSize
The maximum message size as included in the message. This data is not used by the User-based Security module.

securityParameters
These are the security parameters. They will be filled in by the User-based Security module.

securityModel
The securityModel in use. Should be User-based Security Model. This data is not used by the User-based Security module.

securityName
Together with the snmpEngineID it identifies a row in the usmUserTablethat is to be used for securing the message. The securityName has a format that is independent of the Security Model. In case of a response this parameter is ignored and the value from the cache is used.

securityLevel
The Level of Security from which the User-based Security module determines if the message needs to be protected from disclosure and if the message needs to be authenticated.

securityEngineID
The snmpEngineID of the authoritative SNMP engine to which a dateRequest message is to be sent. In case of a response it is implied to be the processing SNMP engine's snmpEngineID and so if it is specified, then it is ignored.

scopedPDU
The message payload. The data is opaque as far as the User-based Security Model is concerned.

securityStateReference
A handle/reference to cachedSecurityData to be used when securing an outgoing Response message. This is the exact same handle/reference as it was generated by the User-based Security module when processing the incoming Request message to which this is the Response message.

wholeMsg
The fully encoded and secured message ready for sending on the wire.

wholeMsgLength
The length of the encoded and secured message (wholeMsg).

Upon completion of the process, the User-based Security module returns statusInformation. If the process was successful, the completed message with privacy and authentication applied if such was requested by the specified securityLevel is returned. If the process was not successful, then an errorIndication is returned.

2.5.2. Services for Processing an Incoming SNMP Message

When the Message Processing (MP) Subsystem invokes the User-based Security module to verify proper security of an incoming message, it must use the service provided for an incoming message. The abstract service primitive is:

```
statusInformation =            -- errorIndication or success
                         -- error counter OID/value if error
  processIncomingMsg(
  IN   messageProcessingModel   -- typically, SNMP version
  IN   maxMessageSize           -- of the sending SNMP entity
  IN   securityParameters       -- for the received message
  IN   securityModel            -- for the received message
  IN   securityLevel            -- Level of Security
  IN   wholeMsg                 -- as received on the wire
  IN   wholeMsgLength           -- length as received on the wire
  OUT  securityEngineID         -- authoritative SNMP entity
  OUT  securityName             -- identification of the principal
  OUT  scopedPDU,               -- message (plaintext) payload
  OUT  maxSizeResponseScopedPDU -- maximum size of the Response PDU
  OUT  securityStateReference   -- reference to security state
      )                         -- information, needed for response
```

The abstract data elements passed as parameters in the abstract service primitives are as follows:

statusInformation
An indication of whether the process was successful or not. If not, then the statusInformation includes the OID and the value of the error counter that was incremented.

messageProcessingModel
The SNMP version number as received in the message. This data is not used by the User-based Security module.

maxMessageSize
The maximum message size as included in the message. The User-bas User-based Security module uses this value to calculate the maxSizeResponseScopedPDU.

securityParameters
These are the security parameters as received in the message.

securityModel
The securityModel in use. Should be the User-based Security Model. This data is not used by the User-based Security module.

securityLevel
The Level of Security from which the User-based Security module determines if the message needs to be protected from disclosure and if the message needs to be authenticated.

wholeMsg
The whole message as it was received.

wholeMsgLength
The length of the message as it was received (wholeMsg).

securityEngineID
The snmpEngineID that was extracted from the field msgAuthoritativeEngineID and that was used to lookup the secrets in the usmUserTable.

securityName
The security name representing the user on whose behalf the message was received. The securityName has a format that is independent of the Security Model.

scopedPDU

The message payload. The data is opaque as far as the User-based Security Model is concerned.

maxSizeResponseScopedPDU

The maximum size of a scopedPDU to be included in a possible Response message. The User-based Security module calculates this size based on the msgMaxSize (as received in the message) and the space required for the message header (including the securityParameters) for such a Response message.

securityStateReference

A handle/reference to cachedSecurityData to be used when securing an outgoing Response message. When the Message Processing Subsystem calls the User-based Security module to generate a

response to this incoming message it must pass this handle/reference.

Upon completion of the process, the User-based Security module returns statusInformation and, if the process was successful, the additional data elements for further processing of the message. If the process was not successful, then an errorIndication, possibly with a OID and value pair of an error counter that was incremented.

2.6. Key Localization Algorithm.

A localized key is a secret key shared between a user U and one authoritative SNMP engine E. Even though a user may have only one password and therefore one key for the whole network, the actual secrets shared between the user and each authoritative SNMP engine will be different. This is achieved by key localization [Localized-key].

First, if a user uses a password, then the user's password is converted into a key Ku using one of the two algorithms described in Appendices A.2.1 and A.2.2.

To convert key Ku into a localized key Kul of user U at the authoritative SNMP engine E, one appends the snmpEngineID of the authoritative SNMP engine to the key Ku and then appends the key Ku to the result, thus enveloping the snmpEngineID within the two copies of user's key Ku. Then one runs a secure hash function (which one depends on the authentication protocol defined for this user U at authoritative SNMP engine E; this document defines two authentication protocols with their associated algorithms based on MD5 and SHA). The output of the hash-function is the localized key Kul for user U at the authoritative SNMP engine E.

3. Elements of Procedure

This section describes the security related procedures followed by an SNMP engine when processing SNMP messages according to the User-based Security Model.

3.1. Generating an Outgoing SNMP Message

This section describes the procedure followed by an SNMP engine whenever it generates a message containing a management operation (like a request, a response, a notification, or a report) on behalf of a user, with a particular securityLevel.

1) a) If any securityStateReference is passed (Response or Report message), then information concerning the user is extracted from the cachedSecurityData. The cachedSecurityData can now be discarded. The securityEngineID is set to the local snmpEngineID. The securityLevel is set to the value specified by the calling module.

Otherwise,

b) based on the securityName, information concerning the user at the destination snmpEngineID, specified by the securityEngineID, is extracted from the Local Configuration Datastore (LCD, usmUserTable). If information about the user is absent from the LCD, then an error indication (unknownSecurityName) is returned to the calling module.

2) If the securityLevel specifies that the message is to be protected from disclosure, but the user does not support both an authentication and a privacy protocol then the message cannot be sent. An error indication (unsupportedSecurityLevel) is returned to the calling module.

3) If the securityLevel specifies that the message is to be authenticated, but the user does not support an authentication protocol, then the message cannot be sent. An error indication (unsupportedSecurityLevel) is returned to the calling module.

4) a) If the securityLevel specifies that the message is to be protected from disclosure, then the octet sequence representing the serialized scopedPDU is encrypted according to the user's privacy protocol. To do so a call is made to the privacy module that implements the user's privacy protocol according to the abstract primitive:

```
statusInformation =       -- success or failure
  encryptData(
  IN    encryptKey        -- user's localized privKey
  IN    dataToEncrypt     -- serialized scopedPDU
  OUT   encryptedData     -- serialized encryptedPDU
  OUT   privParameters    -- serialized privacy parameters
       )
```

statusInformation
  indicates if the encryption process was successful or not.

encryptKey
  the user's localized private privKey is the secret key that can be used by the encryption algorithm.

dataToEncrypt
  the serialized scopedPDU is the data to be encrypted.

encryptedData
  the encryptedPDU represents the encrypted scopedPDU, encoded as an OCTET STRING.

privParameters

the privacy parameters, encoded as an OCTET STRING.

If the privacy module returns failure, then the message cannot be sent and an error indication (encryptionError) is returned to the calling module.

If the privacy module returns success, then the returned privParameters are put into the msgPrivacyParameters field of the securityParameters and the encryptedPDU serves as the payload of the message being prepared.

Otherwise,

b) If the securityLevel specifies that the message is not to be be protected from disclosure, then a zero-length OCTET STRING is encoded into the msgPrivacyParameters field of the securityParameters and the plaintext scopedPDU serves as the payload of the message being prepared.

5) The securityEngineID is encoded as an OCTET STRING into the msgAuthoritativeEngineID field of the securityParameters. Note that an empty (zero length) securityEngineID is OK for a Request message, because that will cause the remote (authoritative) SNMP engine to return a Report PDU with the proper securityEngineID included in the msgAuthoritativeEngineID in the securityParameters of that returned Report PDU.

6) a) If the securityLevel specifies that the message is to be authenticated, then the current values of snmpEngineBoots and snmpEngineTime corresponding to the securityEngineID from the LCD are used.

Otherwise,

b) If this is a Response or Report message, then the current value of snmpEngineBoots and snmpEngineTime corresponding to the local snmpEngineID from the LCD are used.

Otherwise,

c) If this is a Request message, then a zero value is used for both snmpEngineBoots and snmpEngineTime. This zero value gets used if snmpEngineID is empty.

The values are encoded as INTEGER respectively into the msgAuthoritativeEngineBoots and msgAuthoritativeEngineTime fields of the securityParameters.

7) The userName is encoded as an OCTET STRING into the msgUserName field of the securityParameters.

8) a) If the securityLevel specifies that the message is to be authenticated, the message is authenticated according to the user's authentication protocol. To do so a call is made to the authentication module that implements the user's authentication protocol according to the abstract service primitive:

```
statusInformation =
  authenticateOutgoingMsg(
  IN  authKey              -- the user's localized authKey
  IN  wholeMsg             -- unauthenticated message
  OUT authenticatedWholeMsg -- authenticated complete message
     )
```

statusInformation
  indicates if authentication was successful or not.

authKey
  the user's localized private authKey is the secret key that can be used by the authentication algorithm.

wholeMsg
  the complete serialized message to be authenticated.

authenticatedWholeMsg
  the same as the input given to the authenticateOutgoingMsg service, but with msgAuthenticationParameters properly filled in.

If the authentication module returns failure, then the message cannot be sent and an error indication (authenticationFailure) is returned to the calling module.

If the authentication module returns success, then the msgAuthenticationParameters field is put into the securityParameters and the authenticatedWholeMsg represents the serialization of the authenticated message being prepared.

Otherwise,

b) If the securityLevel specifies that the message is not to be authenticated then a zero-length OCTET STRING is encoded into the msgAuthenticationParameters field of the securityParameters. The wholeMsg is now serialized and then represents the unauthenticated message being prepared.

9) The completed message with its length is returned to the calling module with the statusInformation set to success.

3.2. Processing an Incoming SNMP Message

This section describes the procedure followed by an SNMP engine whenever it receives a message containing a management operation on behalf of a user, with a particular securityLevel.

To simplify the elements of procedure, the release of state information is not always explicitly specified. As a general rule, if state information is available when a message gets discarded, the state information should also be released. Also, an error indication can return an OID and value for an incremented counter and optionally a value for securityLevel, and values for contextEngineID or contextName for the counter. In addition, the securityStateReference data is returned if any such information is available at the point where the error is detected.

1) If the received securityParameters is not the serialization (according to the conventions of [RFC3417]) of an OCTET STRING formatted according to the UsmSecurityParameters defined in section 2.4, then the snmpInASNParseErrs counter [RFC3418] is incremented, and an error indication (parseError) is returned to the calling module. Note that we return without the OID and value of the incremented counter, because in this case there is not enough information to generate a Report PDU.

2) The values of the security parameter fields are extracted from the securityParameters. The securityEngineID to be returned to the caller is the value of the msgAuthoritativeEngineID field. The cachedSecurityData is prepared and a securityStateReference is prepared to reference this data. Values to be cached are:

msgUserName

3) If the value of the msgAuthoritativeEngineID field in the securityParameters is unknown then:

a) a non-authoritative SNMP engine that performs discovery may optionally create a new entry in its Local Configuration Datastore (LCD) and continue processing;

or

b) the usmStatsUnknownEngineIDs counter is incremented, and an error indication (unknownEngineID) together with the OID and value of the incremented counter is returned to the calling module.

Note in the event that a zero-length, or other illegally sized msgAuthoritativeEngineID is received, b) should be chosen to facilitate engineID discovery. Otherwise the choice between a) and b) is an implementation issue.

4) Information about the value of the msgUserName and msgAuthoritativeEngineID fields is extracted from the Local Configuration Datastore (LCD, usmUserTable). If no information is available for the user, then the usmStatsUnknownUserNames counter is incremented and an error indication (unknownSecurityName) together with the OID and value of the incremented counter is returned to the calling module.

5) If the information about the user indicates that it does not support the securityLevel requested by the caller, then the usmStatsUnsupportedSecLevels counter is incremented and an error indication (unsupportedSecurityLevel) together with the OID and value of the incremented counter is returned to the calling

module.

6) If the securityLevel specifies that the message is to be authenticated, then the message is authenticated according to the user's authentication protocol. To do so a call is made to the authentication module that implements the user's authentication protocol according to the abstract service primitive:

```
statusInformation =          -- success or failure
  authenticateIncomingMsg(
  IN   authKey               -- the user's localized authKey
  IN   authParameters        -- as received on the wire
  IN   wholeMsg              -- as received on the wire
  OUT  authenticatedWholeMsg -- checked for authentication
       )
```


statusInformation
  indicates if authentication was successful or not.

authKey
  the user's localized private authKey is the secret key that can be used by the authentication algorithm.

wholeMsg
  the complete serialized message to be authenticated.

authenticatedWholeMsg
  the same as the input given to the authenticateIncomingMsg service, but after authentication has been checked.

If the authentication module returns failure, then the message cannot be trusted, so the usmStatsWrongDigests counter is incremented and an error indication (authenticationFailure) together with the OID and value of the incremented counter is returned to the calling module.

If the authentication module returns success, then the message is authentic and can be trusted so processing continues.

7) If the securityLevel indicates an authenticated message, then the

local values of snmpEngineBoots, snmpEngineTime and latestReceivedEngineTime corresponding to the value of the msgAuthoritativeEngineID field are extracted from the Local Configuration Datastore.

a) If the extracted value of msgAuthoritativeEngineID is the same as the value of snmpEngineID of the processing SNMP engine (meaning this is the authoritative SNMP engine), then if any of the following conditions is true, then the message is considered to be outside of the Time Window:

- the local value of snmpEngineBoots is 2147483647;

- the value of the msgAuthoritativeEngineBoots field differs from the local value of snmpEngineBoots; or,

- the value of the msgAuthoritativeEngineTime field differs from the local notion of snmpEngineTime by more than +/- 150 seconds.

If the message is considered to be outside of the Time Window then the usmStatsNotInTimeWindows counter is incremented and an error indication (notInTimeWindow) together with the OID, the value of the incremented counter, and an indication that

the error must be reported with a securityLevel of authNoPriv, is returned to the calling module

b) If the extracted value of msgAuthoritativeEngineID is not the same as the value snmpEngineID of the processing SNMP engine (meaning this is not the authoritative SNMP engine), then:

1) if at least one of the following conditions is true:

- the extracted value of the msgAuthoritativeEngineBoots field is greater than the local notion of the value of snmpEngineBoots; or,

- the extracted value of the msgAuthoritativeEngineBoots field is equal to the local notion of the value of snmpEngineBoots, and the extracted value of msgAuthoritativeEngineTime field is greater than the value of latestReceivedEngineTime,

then the LCD entry corresponding to the extracted value of the msgAuthoritativeEngineID field is updated, by setting:

- the local notion of the value of snmpEngineBoots to the value of the msgAuthoritativeEngineBoots field,

- the local notion of the value of snmpEngineTime to the value of the msgAuthoritativeEngineTime field, and

- the latestReceivedEngineTime to the value of the value of the msgAuthoritativeEngineTime field.

2) if any of the following conditions is true, then the message is considered to be outside of the Time Window:

- the local notion of the value of snmpEngineBoots is 2147483647;

- the value of the msgAuthoritativeEngineBoots field is less than the local notion of the value of snmpEngineBoots; or,

- the value of the msgAuthoritativeEngineBoots field is equal to the local notion of the value of snmpEngineBoots and the value of the msgAuthoritativeEngineTime field is more than 150 seconds less than the local notion of the value of snmpEngineTime.

If the message is considered to be outside of the Time Window then an error indication (notInTimeWindow) is returned to the calling module.

Note that this means that a too old (possibly replayed) message has been detected and is deemed unauthentic.

Note that this procedure allows for the value of msgAuthoritativeEngineBoots in the message to be greater than the local notion of the value of snmpEngineBoots to allow for received messages to be accepted as authentic when received from an authoritative SNMP engine that has re-booted since the receiving SNMP engine last (re-)synchronized.

8) a) If the securityLevel indicates that the message was protected from disclosure, then the OCTET STRING representing the encryptedPDU is decrypted according to the user's privacy protocol to obtain an unencrypted serialized scopedPDU value. To do so a call is made to the privacy module that implements the user's privacy protocol according to the abstract primitive:

```
statusInformation =       -- success or failure
  decryptData(
  IN    decryptKey        -- the user's localized privKey
  IN    privParameters    -- as received on the wire
  IN    encryptedData     -- encryptedPDU as received
  OUT   decryptedData     -- serialized decrypted scopedPDU
       )
```

statusInformation
indicates if the decryption process was successful or not.

decryptKey
the user's localized private privKey is the secret key that can be used by the decryption algorithm.

privParameters
the msgPrivacyParameters, encoded as an OCTET STRING.

encryptedData
the encryptedPDU represents the encrypted scopedPDU, encoded as an OCTET STRING.

decryptedData
the serialized scopedPDU if decryption is successful.

If the privacy module returns failure, then the message can not be processed, so the usmStatsDecryptionErrors counter is incremented and an error indication (decryptionError) together with the OID and value of the incremented counter is returned to the calling module.

If the privacy module returns success, then the decrypted scopedPDU is the message payload to be returned to the calling module.

Otherwise,

b) The scopedPDU component is assumed to be in plain text and is the message payload to be returned to the calling module.

9) The maxSizeResponseScopedPDU is calculated. This is the maximum size allowed for a scopedPDU for a possible Response message. Provision is made for a message header that allows the same securityLevel as the received Request.

10) The securityName for the user is retrieved from the usmUserTable.

11) The security data is cached as cachedSecurityData, so that a possible response to this message can and will use the same authentication and privacy secrets. Information to be saved/cached is as follows:

msgUserName,
usmUserAuthProtocol, usmUserAuthKey
usmUserPrivProtocol, usmUserPrivKey

12) The statusInformation is set to success and a return is made to the calling module passing back the OUT parameters as specified in the processIncomingMsg primitive.

4. Discovery

The User-based Security Model requires that a discovery process obtains sufficient information about other SNMP engines in order to communicate with them. Discovery requires an non-authoritative SNMP engine to learn the authoritative SNMP engine's snmpEngineID value before communication may proceed. This may be accomplished by generating a Request message with a securityLevel of noAuthNoPriv, a msgUserName of zero-length, a msgAuthoritativeEngineID value of zero length, and the varBindList left empty. The response to this message will be a Report message containing the snmpEngineID of the authoritative SNMP engine as the value of the msgAuthoritativeEngineID field within the msgSecurityParameters field. It contains a Report PDU with the usmStatsUnknownEngineIDs counter in the varBindList.

If authenticated communication is required, then the discovery process should also establish time synchronization with the authoritative SNMP engine. This may be accomplished by sending an authenticated Request message with the value of msgAuthoritativeEngineID set to the newly learned snmpEngineID and with the values of msgAuthoritativeEngineBoots and msgAuthoritativeEngineTime set to zero. For an authenticated Request message, a valid userName must be used in the msgUserName field. The response to this authenticated message will be a Report message containing the up to date values of the authoritative SNMP engine's snmpEngineBoots and snmpEngineTime as the value of the msgAuthoritativeEngineBoots and msgAuthoritativeEngineTime fields respectively. It also contains the usmStatsNotInTimeWindows counter in the varBindList of the Report PDU. The time synchronization then happens automatically as part of the procedures in section 3.2 step 7b. See also section 2.3.

5. Definitions

```
SNMP-USER-BASED-SM-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE,
   OBJECT-IDENTITY,
   snmpModules, Counter32                FROM SNMPv2-SMI
   TEXTUAL-CONVENTION, TestAndIncr,
   RowStatus, RowPointer,
   StorageType, AutonomousType           FROM SNMPv2-TC
   MODULE-COMPLIANCE, OBJECT-GROUP       FROM SNMPv2-CONF
   SnmpAdminString, SnmpEngineID,
   snmpAuthProtocols, snmpPrivProtocols  FROM SNMP-FRAMEWORK-MIB;

snmpUsmMIB MODULE-IDENTITY
   LAST-UPDATED "200210160000Z"          -- 16 Oct 2002, midnight
   ORGANIZATION "SNMPv3 Working Group"
   CONTACT-INFO "WG-email:   snmpv3@lists.tislabs.com
                 Subscribe:  majordomo@lists.tislabs.com
```

In msg body: subscribe snmpv3

Chair: Russ Mundy
Network Associates Laboratories
postal: 15204 Omega Drive, Suite 300
Rockville, MD 20850-4601
USA
email: mundy@tislabs.com

phone: +1 301-947-7107

Co-Chair: David Harrington
Enterasys Networks
Postal: 35 Industrial Way
P. O. Box 5004
Rochester, New Hampshire 03866-5005
USA
EMail: dbh@enterasys.com
Phone: +1 603-337-2614

Co-editor   Uri Blumenthal
Lucent Technologies
postal: 67 Whippany Rd.
Whippany, NJ 07981
USA
email: uri@lucent.com
phone: +1-973-386-2163

Co-editor: Bert Wijnen
Lucent Technologies
postal: Schagen 33
3461 GL Linschoten
Netherlands
email: bwijnen@lucent.com
phone: +31-348-480-685
"

DESCRIPTION "The management information definitions for the
SNMP User-based Security Model.

Copyright (C) The Internet Society (2002). This
version of this MIB module is part of RFC 3414;

```
              see the RFC itself for full legal notices.
             "
--  Revision history

   REVISION     "200210160000Z"          -- 16 Oct 2002, midnight
   DESCRIPTION  "Changes in this revision:
              - Updated references and contact info.
              - Clarification to usmUserCloneFrom DESCRIPTION
                clause
              - Fixed 'command responder' into 'command generator'
                in last para of DESCRIPTION clause of
                usmUserTable.
              This revision published as RFC3414.
             "
   REVISION     "199901200000Z"          -- 20 Jan 1999, midnight
   DESCRIPTION  "Clarifications, published as RFC2574"
```



```
   REVISION     "199711200000Z"          -- 20 Nov 1997, midnight
   DESCRIPTION  "Initial version, published as RFC2274"

   ::= { snmpModules 15 }

-- Administrative assignments ****************************************

usmMIBObjects     OBJECT IDENTIFIER ::= { snmpUsmMIB 1 }
usmMIBConformance OBJECT IDENTIFIER ::= { snmpUsmMIB 2 }

-- Identification of Authentication and Privacy Protocols ************

usmNoAuthProtocol OBJECT-IDENTITY
   STATUS        current
   DESCRIPTION  "No Authentication Protocol."
   ::= { snmpAuthProtocols 1 }

usmHMACMD5AuthProtocol OBJECT-IDENTITY
   STATUS        current
   DESCRIPTION  "The HMAC-MD5-96 Digest Authentication Protocol."
   REFERENCE    "- H. Krawczyk, M. Bellare, R. Canetti HMAC:
                Keyed-Hashing for Message Authentication,
                RFC2104, Feb 1997.
              - Rivest, R., Message Digest Algorithm MD5, RFC1321.
```

```
            "
  ::= { snmpAuthProtocols 2 }

usmHMACSHAAuthProtocol OBJECT-IDENTITY
  STATUS        current
  DESCRIPTION  "The HMAC-SHA-96 Digest Authentication Protocol."
  REFERENCE    "- H. Krawczyk, M. Bellare, R. Canetti, HMAC:
                Keyed-Hashing for Message Authentication,
                RFC2104, Feb 1997.
              - Secure Hash Algorithm. NIST FIPS 180-1.
             "
  ::= { snmpAuthProtocols 3 }

usmNoPrivProtocol OBJECT-IDENTITY
  STATUS        current
  DESCRIPTION  "No Privacy Protocol."
  ::= { snmpPrivProtocols 1 }

usmDESPrivProtocol OBJECT-IDENTITY
  STATUS        current
  DESCRIPTION  "The CBC-DES Symmetric Encryption Protocol."
  REFERENCE    "- Data Encryption Standard, National Institute of
                Standards and Technology.  Federal Information
                Processing Standard (FIPS) Publication 46-1.
```



```
                Supersedes FIPS Publication 46,
                (January, 1977; reaffirmed January, 1988).

              - Data Encryption Algorithm, American National
                Standards Institute.  ANSI X3.92-1981,
                (December, 1980).

              - DES Modes of Operation, National Institute of
                Standards and Technology.  Federal Information
                Processing Standard (FIPS) Publication 81,
                (December, 1980).

              - Data Encryption Algorithm - Modes of Operation,
                American National Standards Institute.
                ANSI X3.106-1983, (May 1983).
             "
```

::= { snmpPrivProtocols 2 }

-- Textual Conventions ***

KeyChange ::= TEXTUAL-CONVENTION
 STATUS current
 DESCRIPTION
 "Every definition of an object with this syntax must identify
 a protocol P, a secret key K, and a hash algorithm H
 that produces output of L octets.

 The object's value is a manager-generated, partially-random
 value which, when modified, causes the value of the secret
 key K, to be modified via a one-way function.

 The value of an instance of this object is the concatenation
 of two components: first a 'random' component and then a
 'delta' component.

 The lengths of the random and delta components
 are given by the corresponding value of the protocol P;
 if P requires K to be a fixed length, the length of both the
 random and delta components is that fixed length; if P
 allows the length of K to be variable up to a particular
 maximum length, the length of the random component is that
 maximum length and the length of the delta component is any
 length less than or equal to that maximum length.
 For example, usmHMACMD5AuthProtocol requires K to be a fixed
 length of 16 octets and L - of 16 octets.
 usmHMACSHAAuthProtocol requires K to be a fixed length of
 20 octets and L - of 20 octets. Other protocols may define
 other sizes, as deemed appropriate.

Blumenthal & Wijnen Standards Track [Page 35]

RFC 3414 USM for SNMPv3 December 2002

 When a requester wants to change the old key K to a new
 key keyNew on a remote entity, the 'random' component is
 obtained from either a true random generator, or from a
 pseudorandom generator, and the 'delta' component is
 computed as follows:

 - a temporary variable is initialized to the existing value
 of K;

- if the length of the keyNew is greater than L octets,
  then:
  - the random component is appended to the value of the
    temporary variable, and the result is input to the
    the hash algorithm H to produce a digest value, and
    the temporary variable is set to this digest value;
  - the value of the temporary variable is XOR-ed with
    the first (next) L-octets (16 octets in case of MD5)
    of the keyNew to produce the first (next) L-octets
    (16 octets in case of MD5) of the 'delta' component.
  - the above two steps are repeated until the unused
    portion of the keyNew component is L octets or less,
- the random component is appended to the value of the
  temporary variable, and the result is input to the
  hash algorithm H to produce a digest value;
- this digest value, truncated if necessary to be the same
  length as the unused portion of the keyNew, is XOR-ed
  with the unused portion of the keyNew to produce the
  (final portion of the) 'delta' component.

For example, using MD5 as the hash algorithm H:

```
iterations = (lenOfDelta - 1)/16; /* integer division */
temp = keyOld;
for (i = 0; i < iterations; i++) {
    temp = MD5 (temp || random);
    delta[i*16 .. (i*16)+15] =
           temp XOR keyNew[i*16 .. (i*16)+15];
}
temp = MD5 (temp || random);
delta[i*16 .. lenOfDelta-1] =
       temp XOR keyNew[i*16 .. lenOfDelta-1];
```

The 'random' and 'delta' components are then concatenated as
described above, and the resulting octet string is sent to
the recipient as the new value of an instance of this object.

At the receiver side, when an instance of this object is set
to a new value, then a new value of K is computed as follows:

- a temporary variable is initialized to the existing value
  of K;
- if the length of the delta component is greater than L
  octets, then:
   - the random component is appended to the value of the
     temporary variable, and the result is input to the
     hash algorithm H to produce a digest value, and the
     temporary variable is set to this digest value;
   - the value of the temporary variable is XOR-ed with
     the first (next) L-octets (16 octets in case of MD5)
     of the delta component to produce the first (next)
     L-octets (16 octets in case of MD5) of the new value
     of K.
   - the above two steps are repeated until the unused
     portion of the delta component is L octets or less,
- the random component is appended to the value of the
  temporary variable, and the result is input to the
  hash algorithm H to produce a digest value;
- this digest value, truncated if necessary to be the same
  length as the unused portion of the delta component, is
  XOR-ed with the unused portion of the delta component to
  produce the (final portion of the) new value of K.

For example, using MD5 as the hash algorithm H:

```
iterations = (lenOfDelta - 1)/16; /* integer division */
temp = keyOld;
for (i = 0; i < iterations; i++) {
    temp = MD5 (temp || random);
    keyNew[i*16 .. (i*16)+15] =
           temp XOR delta[i*16 .. (i*16)+15];
}
temp = MD5 (temp || random);
keyNew[i*16 .. lenOfDelta-1] =
       temp XOR delta[i*16 .. lenOfDelta-1];
```

The value of an object with this syntax, whenever it is retrieved by the management protocol, is always the zero length string.

Note that the keyOld and keyNew are the localized keys.

Note that it is probably wise that when an SNMP entity sends a SetRequest to change a key, that it keeps a copy of the old key until it has confirmed that the key change actually succeeded.
"

SYNTAX OCTET STRING

```
-- Statistics for the User-based Security Model ***********************

usmStats     OBJECT IDENTIFIER ::= { usmMIBObjects 1 }

usmStatsUnsupportedSecLevels OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS  read-only
  STATUS     current
  DESCRIPTION "The total number of packets received by the SNMP
         engine which were dropped because they requested a
         securityLevel that was unknown to the SNMP engine
         or otherwise unavailable.
        "
  ::= { usmStats 1 }

usmStatsNotInTimeWindows OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS  read-only
  STATUS     current
  DESCRIPTION "The total number of packets received by the SNMP
         engine which were dropped because they appeared
         outside of the authoritative SNMP engine's window.
        "
  ::= { usmStats 2 }

usmStatsUnknownUserNames OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS  read-only
  STATUS     current
  DESCRIPTION "The total number of packets received by the SNMP
         engine which were dropped because they referenced a
         user that was not known to the SNMP engine.
        "
  ::= { usmStats 3 }

usmStatsUnknownEngineIDs OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS  read-only
```

```
    STATUS       current
    DESCRIPTION "The total number of packets received by the SNMP
                 engine which were dropped because they referenced an
                 snmpEngineID that was not known to the SNMP engine.
                "
    ::= { usmStats 4 }

usmStatsWrongDigests OBJECT-TYPE
```



```
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION "The total number of packets received by the SNMP
                 engine which were dropped because they didn't
                 contain the expected digest value.
                "
    ::= { usmStats 5 }

usmStatsDecryptionErrors OBJECT-TYPE
    SYNTAX       Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION "The total number of packets received by the SNMP
                 engine which were dropped because they could not be
                 decrypted.
                "
    ::= { usmStats 6 }

-- The usmUser Group ********************************************

usmUser          OBJECT IDENTIFIER ::= { usmMIBObjects 2 }

usmUserSpinLock  OBJECT-TYPE
    SYNTAX       TestAndIncr
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION "An advisory lock used to allow several cooperating
                 Command Generator Applications to coordinate their
                 use of facilities to alter secrets in the
                 usmUserTable.
                "
```

```
  ::= { usmUser 1 }

-- The table of valid users for the User-based Security Model ********

usmUserTable     OBJECT-TYPE
  SYNTAX       SEQUENCE OF UsmUserEntry
  MAX-ACCESS   not-accessible
  STATUS       current
  DESCRIPTION "The table of users configured in the SNMP engine's
              Local Configuration Datastore (LCD).

              To create a new user (i.e., to instantiate a new
              conceptual row in this table), it is recommended to
              follow this procedure:

                1)  GET(usmUserSpinLock.0) and save in sValue.
```

Blumenthal & Wijnen   Standards Track   [Page 39]

RFC 3414   USM for SNMPv3   December 2002

```
                2)  SET(usmUserSpinLock.0=sValue,
                        usmUserCloneFrom=templateUser,
                        usmUserStatus=createAndWait)
                    You should use a template user to clone from
                    which has the proper auth/priv protocol defined.

              If the new user is to use privacy:

                3)  generate the keyChange value based on the secret
                    privKey of the clone-from user and the secret key
                    to be used for the new user. Let us call this
                    pkcValue.
                4)  GET(usmUserSpinLock.0) and save in sValue.
                5)  SET(usmUserSpinLock.0=sValue,
                        usmUserPrivKeyChange=pkcValue
                        usmUserPublic=randomValue1)
                6)  GET(usmUserPulic) and check it has randomValue1.
                    If not, repeat steps 4-6.

              If the new user will never use privacy:

                7)  SET(usmUserPrivProtocol=usmNoPrivProtocol)

              If the new user is to use authentication:
```

8) generate the keyChange value based on the secret
   authKey of the clone-from user and the secret key
   to be used for the new user. Let us call this
   akcValue.
9) GET(usmUserSpinLock.0) and save in sValue.
10) SET(usmUserSpinLock.0=sValue,
      usmUserAuthKeyChange=akcValue
      usmUserPublic=randomValue2)
11) GET(usmUserPulic) and check it has randomValue2.
   If not, repeat steps 9-11.

If the new user will never use authentication:

12) SET(usmUserAuthProtocol=usmNoAuthProtocol)

Finally, activate the new user:

13) SET(usmUserStatus=active)

The new user should now be available and ready to be used for SNMPv3 communication. Note however that access to MIB data must be provided via configuration of the SNMP-VIEW-BASED-ACM-MIB.

The use of usmUserSpinlock is to avoid conflicts with another SNMP command generator application which may also be acting on the usmUserTable.
"
::= { usmUser 2 }

usmUserEntry OBJECT-TYPE
SYNTAX UsmUserEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A user configured in the SNMP engine's Local
Configuration Datastore (LCD) for the User-based
Security Model.
"
INDEX { usmUserEngineID,
usmUserName

```
        }
    ::= { usmUserTable 1 }

UsmUserEntry ::= SEQUENCE
    {
        usmUserEngineID         SnmpEngineID,
        usmUserName             SnmpAdminString,
        usmUserSecurityName     SnmpAdminString,
        usmUserCloneFrom        RowPointer,
        usmUserAuthProtocol     AutonomousType,
        usmUserAuthKeyChange    KeyChange,
        usmUserOwnAuthKeyChange KeyChange,
        usmUserPrivProtocol     AutonomousType,
        usmUserPrivKeyChange    KeyChange,
        usmUserOwnPrivKeyChange KeyChange,
        usmUserPublic           OCTET STRING,
        usmUserStorageType      StorageType,
        usmUserStatus           RowStatus
    }

usmUserEngineID  OBJECT-TYPE
    SYNTAX       SnmpEngineID
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION "An SNMP engine's administratively-unique identifier.

                 In a simple agent, this value is always that agent's
                 own snmpEngineID value.

                 The value can also take the value of the snmpEngineID
                 of a remote SNMP engine with which this user can
                 communicate.
```



```
                "
    ::= { usmUserEntry 1 }

usmUserName      OBJECT-TYPE
    SYNTAX       SnmpAdminString (SIZE(1..32))
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION "A human readable string representing the name of
```

```
            the user.

            This is the (User-based Security) Model dependent
            security ID.
           "
    ::= { usmUserEntry 2 }

usmUserSecurityName OBJECT-TYPE
    SYNTAX       SnmpAdminString
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION "A human readable string representing the user in
            Security Model independent format.

            The default transformation of the User-based Security
            Model dependent security ID to the securityName and
            vice versa is the identity function so that the
            securityName is the same as the userName.
           "
    ::= { usmUserEntry 3 }

usmUserCloneFrom OBJECT-TYPE
    SYNTAX       RowPointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "A pointer to another conceptual row in this
            usmUserTable.  The user in this other conceptual
            row is called the clone-from user.

            When a new user is created (i.e., a new conceptual
            row is instantiated in this table), the privacy and
            authentication parameters of the new user must be
            cloned from its clone-from user. These parameters are:
              - authentication protocol (usmUserAuthProtocol)
              - privacy protocol (usmUserPrivProtocol)
            They will be copied regardless of what the current
            value is.

            Cloning also causes the initial values of the secret
            authentication key (authKey) and the secret encryption
```

```
                key (privKey) of the new user to be set to the same
                values as the corresponding secrets of the clone-from
                user to allow the KeyChange process to occur as
                required during user creation.

                The first time an instance of this object is set by
                a management operation (either at or after its
                instantiation), the cloning process is invoked.
                Subsequent writes are successful but invoke no
                action to be taken by the receiver.
                The cloning process fails with an 'inconsistentName'
                error if the conceptual row representing the
                clone-from user does not exist or is not in an active
                state when the cloning process is invoked.

                When this object is read, the ZeroDotZero OID
                is returned.
               "
    ::= { usmUserEntry 4 }

usmUserAuthProtocol OBJECT-TYPE
    SYNTAX       AutonomousType
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "An indication of whether messages sent on behalf of
                this user to/from the SNMP engine identified by
                usmUserEngineID, can be authenticated, and if so,
                the type of authentication protocol which is used.

                An instance of this object is created concurrently
                with the creation of any other object instance for
                the same user (i.e., as part of the processing of
                the set operation which creates the first object
                instance in the same conceptual row).

                If an initial set operation (i.e. at row creation time)
                tries to set a value for an unknown or unsupported
                protocol, then a 'wrongValue' error must be returned.

                The value will be overwritten/set when a set operation
                is performed on the corresponding instance of
                usmUserCloneFrom.

                Once instantiated, the value of such an instance of
                this object can only be changed via a set operation to
                the value of the usmNoAuthProtocol.

                If a set operation tries to change the value of an
```

```
                existing instance of this object to any value other
                than usmNoAuthProtocol, then an 'inconsistentValue'
                error must be returned.

                If a set operation tries to set the value to the
                usmNoAuthProtocol while the usmUserPrivProtocol value
                in the same row is not equal to usmNoPrivProtocol,
                then an 'inconsistentValue' error must be returned.
                That means that an SNMP command generator application
                must first ensure that the usmUserPrivProtocol is set
                to the usmNoPrivProtocol value before it can set
                the usmUserAuthProtocol value to usmNoAuthProtocol.
               "
    DEFVAL      { usmNoAuthProtocol }
    ::= { usmUserEntry 5 }

usmUserAuthKeyChange OBJECT-TYPE
    SYNTAX       KeyChange   -- typically (SIZE (0 | 32)) for HMACMD5
                             -- typically (SIZE (0 | 40)) for HMACSHA
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "An object, which when modified, causes the secret
                authentication key used for messages sent on behalf
                of this user to/from the SNMP engine identified by
                usmUserEngineID, to be modified via a one-way
                function.

                The associated protocol is the usmUserAuthProtocol.
                The associated secret key is the user's secret
                authentication key (authKey). The associated hash
                algorithm is the algorithm used by the user's
                usmUserAuthProtocol.

                When creating a new user, it is an 'inconsistentName'
                error for a set operation to refer to this object
                unless it is previously or concurrently initialized
                through a set operation on the corresponding instance
                of usmUserCloneFrom.

                When the value of the corresponding usmUserAuthProtocol
```

is usmNoAuthProtocol, then a set is successful, but
effectively is a no-op.

When this object is read, the zero-length (empty)
string is returned.

The recommended way to do a key change is as follows:


```
  1) GET(usmUserSpinLock.0) and save in sValue.
  2) generate the keyChange value based on the old
     (existing) secret key and the new secret key,
     let us call this kcValue.

If you do the key change on behalf of another user:

  3) SET(usmUserSpinLock.0=sValue,
         usmUserAuthKeyChange=kcValue
         usmUserPublic=randomValue)

If you do the key change for yourself:

  4) SET(usmUserSpinLock.0=sValue,
         usmUserOwnAuthKeyChange=kcValue
         usmUserPublic=randomValue)

If you get a response with error-status of noError,
then the SET succeeded and the new key is active.
If you do not get a response, then you can issue a
GET(usmUserPublic) and check if the value is equal
to the randomValue you did send in the SET. If so, then
the key change succeeded and the new key is active
(probably the response got lost). If not, then the SET
request probably never reached the target and so you
can start over with the procedure above.
"
  DEFVAL      { ''H }    -- the empty string
  ::= { usmUserEntry 6 }

usmUserOwnAuthKeyChange OBJECT-TYPE
  SYNTAX       KeyChange   -- typically (SIZE (0 | 32)) for HMACMD5
```

```
                    -- typically (SIZE (0 | 40)) for HMACSHA
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "Behaves exactly as usmUserAuthKeyChange, with one
                 notable difference: in order for the set operation
                 to succeed, the usmUserName of the operation
                 requester must match the usmUserName that
                 indexes the row which is targeted by this
                 operation.
                 In addition, the USM security model must be
                 used for this operation.

                 The idea here is that access to this column can be
                 public, since it will only allow a user to change
                 his own secret authentication key (authKey).
                 Note that this can only be done once the row is active.
```



```
                 When a set is received and the usmUserName of the
                 requester is not the same as the umsUserName that
                 indexes the row which is targeted by this operation,
                 then a 'noAccess' error must be returned.

                 When a set is received and the security model in use
                 is not USM, then a 'noAccess' error must be returned.
                "
    DEFVAL      { ''H }    -- the empty string
    ::= { usmUserEntry 7 }

usmUserPrivProtocol OBJECT-TYPE
    SYNTAX       AutonomousType
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "An indication of whether messages sent on behalf of
                 this user to/from the SNMP engine identified by
                 usmUserEngineID, can be protected from disclosure,
                 and if so, the type of privacy protocol which is used.

                 An instance of this object is created concurrently
                 with the creation of any other object instance for
                 the same user (i.e., as part of the processing of
                 the set operation which creates the first object
```

instance in the same conceptual row).

If an initial set operation (i.e. at row creation time) tries to set a value for an unknown or unsupported protocol, then a 'wrongValue' error must be returned.

The value will be overwritten/set when a set operation is performed on the corresponding instance of usmUserCloneFrom.

Once instantiated, the value of such an instance of this object can only be changed via a set operation to the value of the usmNoPrivProtocol.

If a set operation tries to change the value of an existing instance of this object to any value other than usmNoPrivProtocol, then an 'inconsistentValue' error must be returned.

Note that if any privacy protocol is used, then you must also use an authentication protocol. In other words, if usmUserPrivProtocol is set to anything else than usmNoPrivProtocol, then the corresponding instance of usmUserAuthProtocol cannot have a value of

usmNoAuthProtocol. If it does, then an 'inconsistentValue' error must be returned.
"
DEFVAL { usmNoPrivProtocol }
::= { usmUserEntry 8 }

usmUserPrivKeyChange OBJECT-TYPE
SYNTAX KeyChange -- typically (SIZE (0 | 32)) for DES
MAX-ACCESS read-create
STATUS current
DESCRIPTION "An object, which when modified, causes the secret encryption key used for messages sent on behalf of this user to/from the SNMP engine identified by usmUserEngineID, to be modified via a one-way function.

```
            The associated protocol is the usmUserPrivProtocol.
            The associated secret key is the user's secret
            privacy key (privKey). The associated hash
            algorithm is the algorithm used by the user's
            usmUserAuthProtocol.

            When creating a new user, it is an 'inconsistentName'
            error for a set operation to refer to this object
            unless it is previously or concurrently initialized
            through a set operation on the corresponding instance
            of usmUserCloneFrom.

            When the value of the corresponding usmUserPrivProtocol
            is usmNoPrivProtocol, then a set is successful, but
            effectively is a no-op.

            When this object is read, the zero-length (empty)
            string is returned.
            See the description clause of usmUserAuthKeyChange for
            a recommended procedure to do a key change.
           "
    DEFVAL      { ''H }    -- the empty string
    ::= { usmUserEntry 9 }

usmUserOwnPrivKeyChange OBJECT-TYPE
    SYNTAX       KeyChange  -- typically (SIZE (0 | 32)) for DES
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "Behaves exactly as usmUserPrivKeyChange, with one
            notable difference: in order for the Set operation
            to succeed, the usmUserName of the operation
            requester must match the usmUserName that indexes
```



```
            the row which is targeted by this operation.
            In addition, the USM security model must be
            used for this operation.

            The idea here is that access to this column can be
            public, since it will only allow a user to change
            his own secret privacy key (privKey).
            Note that this can only be done once the row is active.
```

```
             When a set is received and the usmUserName of the
             requester is not the same as the umsUserName that
             indexes the row which is targeted by this operation,
             then a 'noAccess' error must be returned.

             When a set is received and the security model in use
             is not USM, then a 'noAccess' error must be returned.
            "
    DEFVAL      { ''H }     -- the empty string
    ::= { usmUserEntry 10 }

usmUserPublic    OBJECT-TYPE
    SYNTAX       OCTET STRING (SIZE(0..32))
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "A publicly-readable value which can be written as part
             of the procedure for changing a user's secret
             authentication and/or privacy key, and later read to
             determine whether the change of the secret was
             effected.
            "
    DEFVAL      { ''H }  -- the empty string
    ::= { usmUserEntry 11 }

usmUserStorageType OBJECT-TYPE
    SYNTAX       StorageType
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "The storage type for this conceptual row.

             Conceptual rows having the value 'permanent' must
             allow write-access at a minimum to:

             - usmUserAuthKeyChange, usmUserOwnAuthKeyChange
               and usmUserPublic for a user who employs
               authentication, and
             - usmUserPrivKeyChange, usmUserOwnPrivKeyChange
               and usmUserPublic for a user who employs
               privacy.
```

```
                Note that any user who employs authentication or
                privacy must allow its secret(s) to be updated and
                thus cannot be 'readOnly'.

                If an initial set operation tries to set the value to
                'readOnly' for a user who employs authentication or
                privacy, then an 'inconsistentValue' error must be
                returned.  Note that if the value has been previously
                set (implicit or explicit) to any value, then the rules
                as defined in the StorageType Textual Convention apply.

                It is an implementation issue to decide if a SET for
                a readOnly or permanent row is accepted at all. In some
                contexts this may make sense, in others it may not. If
                a SET for a readOnly or permanent row is not accepted
                at all, then a 'wrongValue' error must be returned.
               "
    DEFVAL      { nonVolatile }
    ::= { usmUserEntry 12 }

usmUserStatus    OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION "The status of this conceptual row.

                Until instances of all corresponding columns are
                appropriately configured, the value of the
                corresponding instance of the usmUserStatus column
                is 'notReady'.

                In particular, a newly created row for a user who
                employs authentication, cannot be made active until the
                corresponding usmUserCloneFrom and usmUserAuthKeyChange
                have been set.

                Further, a newly created row for a user who also
                employs privacy, cannot be made active until the
                usmUserPrivKeyChange has been set.

                The RowStatus TC [RFC2579] requires that this
                DESCRIPTION clause states under which circumstances
                other objects in this row can be modified:

                The value of this object has no effect on whether
                other objects in this conceptual row can be modified,
                except for usmUserOwnAuthKeyChange and
                usmUserOwnPrivKeyChange. For these 2 objects, the
```

Blumenthal & Wijnen Standards Track [Page 49]


```
              value of usmUserStatus MUST be active.
             "
   ::= { usmUserEntry 13 }

-- Conformance Information *******************************************

usmMIBCompliances OBJECT IDENTIFIER ::= { usmMIBConformance 1 }
usmMIBGroups      OBJECT IDENTIFIER ::= { usmMIBConformance 2 }

-- Compliance statements

usmMIBCompliance MODULE-COMPLIANCE
   STATUS       current
   DESCRIPTION "The compliance statement for SNMP engines which
              implement the SNMP-USER-BASED-SM-MIB.
             "

   MODULE       -- this module
      MANDATORY-GROUPS { usmMIBBasicGroup }

      OBJECT           usmUserAuthProtocol
      MIN-ACCESS       read-only
      DESCRIPTION     "Write access is not required."

      OBJECT           usmUserPrivProtocol
      MIN-ACCESS       read-only
      DESCRIPTION     "Write access is not required."

   ::= { usmMIBCompliances 1 }

-- Units of compliance
usmMIBBasicGroup OBJECT-GROUP
   OBJECTS     {
               usmStatsUnsupportedSecLevels,
               usmStatsNotInTimeWindows,
               usmStatsUnknownUserNames,
               usmStatsUnknownEngineIDs,
               usmStatsWrongDigests,
               usmStatsDecryptionErrors,
               usmUserSpinLock,
```

```
            usmUserSecurityName,
            usmUserCloneFrom,
            usmUserAuthProtocol,
            usmUserAuthKeyChange,
            usmUserOwnAuthKeyChange,
            usmUserPrivProtocol,
            usmUserPrivKeyChange,
            usmUserOwnPrivKeyChange,
```



```
            usmUserPublic,
            usmUserStorageType,
            usmUserStatus
           }
  STATUS       current
  DESCRIPTION "A collection of objects providing for configuration
           of an SNMP engine which implements the SNMP
           User-based Security Model.
          "
  ::= { usmMIBGroups 1 }

END
```

6. HMAC-MD5-96 Authentication Protocol

This section describes the HMAC-MD5-96 authentication protocol. This authentication protocol is the first defined for the User-based Security Model. It uses MD5 hash-function which is described in [RFC1321], in HMAC mode described in [RFC2104], truncating the output to 96 bits.

This protocol is identified by usmHMACMD5AuthProtocol.

Over time, other authentication protocols may be defined either as a replacement of this protocol or in addition to this protocol.

6.1. Mechanisms

- In support of data integrity, a message digest algorithm is required. A digest is calculated over an appropriate portion of an SNMP message and included as part of the message sent to the recipient.

- In support of data origin authentication and data integrity, a secret value is prepended to SNMP message prior to computing the digest; the calculated digest is partially inserted into the SNMP message prior to transmission, and the prepended value is not transmitted. The secret value is shared by all SNMP engines authorized to originate messages on behalf of the appropriate user.

6.1.1. Digest Authentication Mechanism

The Digest Authentication Mechanism defined in this memo provides for:

- verification of the integrity of a received message, i.e., the message received is the message sent.

The integrity of the message is protected by computing a digest over an appropriate portion of the message. The digest is computed by the originator of the message, transmitted with the message, and verified by the recipient of the message.

- verification of the user on whose behalf the message was generated.

A secret value known only to SNMP engines authorized to generate messages on behalf of a user is used in HMAC mode (see [RFC2104]). It also recommends the hash-function output used as Message Authentication Code, to be truncated.

This protocol uses the MD5 [RFC1321] message digest algorithm. A 128-bit MD5 digest is calculated in a special (HMAC) way over the designated portion of an SNMP message and the first 96 bits of this digest is included as part of the message sent to the recipient. The size of the digest carried in a message is 12 octets. The size of the private authentication key (the secret) is 16 octets. For the details see section 6.3.

6.2. Elements of the Digest Authentication Protocol

This section contains definitions required to realize the authentication module defined in this section of this memo.

6.2.1. Users

Authentication using this authentication protocol makes use of a defined set of userNames. For any user on whose behalf a message must be authenticated at a particular SNMP engine, that SNMP engine must have knowledge of that user. An SNMP engine that wishes to communicate with another SNMP engine must also have knowledge of a user known to that engine, including knowledge of the applicable attributes of that user.

A user and its attributes are defined as follows:

<userName>
  A string representing the name of the user.
<authKey>
  A user's secret key to be used when calculating a digest.
  It MUST be 16 octets long for MD5.

6.2.2. msgAuthoritativeEngineID

The msgAuthoritativeEngineID value contained in an authenticated message specifies the authoritative SNMP engine for that particular message (see the definition of SnmpEngineID in the SNMP Architecture document [RFC3411]).

The user's (private) authentication key is normally different at each authoritative SNMP engine and so the snmpEngineID is used to select the proper key for the authentication process.

6.2.3. SNMP Messages Using this Authentication Protocol

Messages using this authentication protocol carry a msgAuthenticationParameters field as part of the msgSecurityParameters. For this protocol, the

msgAuthenticationParameters field is the serialized OCTET STRING
representing the first 12 octets of the HMAC-MD5-96 output done over
the wholeMsg.

The digest is calculated over the wholeMsg so if a message is
authenticated, that also means that all the fields in the message are
intact and have not been tampered with.

6.2.4. Services provided by the HMAC-MD5-96 Authentication Module

This section describes the inputs and outputs that the HMAC-MD5-96
Authentication module expects and produces when the User-based
Security module calls the HMAC-MD5-96 Authentication module for
services.

6.2.4.1. Services for Generating an Outgoing SNMP Message

The HMAC-MD5-96 authentication protocol assumes that the selection of
the authKey is done by the caller and that the caller passes the
secret key to be used.

Upon completion the authentication module returns statusInformation
and, if the message digest was correctly calculated, the wholeMsg
with the digest inserted at the proper place. The abstract service
primitive is:

```
statusInformation =              -- success or failure
  authenticateOutgoingMsg(
  IN   authKey                   -- secret key for authentication
  IN   wholeMsg                  -- unauthenticated complete message
  OUT  authenticatedWholeMsg     -- complete authenticated message
      )
```


The abstract data elements are:

statusInformation
  An indication of whether the authentication process was successful.
  If not it is an indication of the problem.

authKey
  The secret key to be used by the authentication algorithm. The

length of this key MUST be 16 octets.

wholeMsg
The message to be authenticated.

authenticatedWholeMsg
The authenticated message (including inserted digest) on output.

Note, that authParameters field is filled by the authentication module and this module and this field should be already present in the wholeMsg before the Message Authentication Code (MAC) is generated.

6.2.4.2. Services for Processing an Incoming SNMP Message

The HMAC-MD5-96 authentication protocol assumes that the selection of the authKey is done by the caller and that the caller passes the secret key to be used.

Upon completion the authentication module returns statusInformation and, if the message digest was correctly calculated, the wholeMsg as it was processed. The abstract service primitive is:

```
statusInformation =              -- success or failure
  authenticateIncomingMsg(
  IN   authKey                   -- secret key for authentication
  IN   authParameters            -- as received on the wire
  IN   wholeMsg                  -- as received on the wire
  OUT  authenticatedWholeMsg     -- complete authenticated message
       )
```

The abstract data elements are:

statusInformation
An indication of whether the authentication process was successful. If not it is an indication of the problem.

authKey
The secret key to be used by the authentication algorithm. The length of this key MUST be 16 octets.

authParameters
The authParameters from the incoming message.

wholeMsg
The message to be authenticated on input and the authenticated message on output.

authenticatedWholeMsg
The whole message after the authentication check is complete.

6.3. Elements of Procedure

This section describes the procedures for the HMAC-MD5-96 authentication protocol.

6.3.1. Processing an Outgoing Message

This section describes the procedure followed by an SNMP engine whenever it must authenticate an outgoing message using the usmHMACMD5AuthProtocol.

1) The msgAuthenticationParameters field is set to the serialization, according to the rules in [RFC3417], of an OCTET STRING containing 12 zero octets.

2) From the secret authKey, two keys K1 and K2 are derived:

   a) extend the authKey to 64 octets by appending 48 zero octets; save it as extendedAuthKey

   b) obtain IPAD by replicating the octet 0x36 64 times;

   c) obtain K1 by XORing extendedAuthKey with IPAD;

   d) obtain OPAD by replicating the octet 0x5C 64 times;

   e) obtain K2 by XORing extendedAuthKey with OPAD.

3) Prepend K1 to the wholeMsg and calculate MD5 digest over it according to [RFC1321].

4) Prepend K2 to the result of the step 4 and calculate MD5 digest over it according to [RFC1321]. Take the first 12 octets of the final digest - this is Message Authentication Code (MAC).

5) Replace the msgAuthenticationParameters field with MAC obtained in the step 4.

6) The authenticatedWholeMsg is then returned to the caller together with statusInformation indicating success.

6.3.2. Processing an Incoming Message

This section describes the procedure followed by an SNMP engine whenever it must authenticate an incoming message using the usmHMACMD5AuthProtocol.

1) If the digest received in the msgAuthenticationParameters field is not 12 octets long, then an failure and an errorIndication (authenticationError) is returned to the calling module.

2) The MAC received in the msgAuthenticationParameters field is saved.

3) The digest in the msgAuthenticationParameters field is replaced by the 12 zero octets.

4) From the secret authKey, two keys K1 and K2 are derived:

   a) extend the authKey to 64 octets by appending 48 zero octets; save it as extendedAuthKey

   b) obtain IPAD by replicating the octet 0x36 64 times;

   c) obtain K1 by XORing extendedAuthKey with IPAD;

   d) obtain OPAD by replicating the octet 0x5C 64 times;

   e) obtain K2 by XORing extendedAuthKey with OPAD.

5) The MAC is calculated over the wholeMsg:

   a) prepend K1 to the wholeMsg and calculate the MD5 digest over it;

   b) prepend K2 to the result of step 5.a and calculate the MD5 digest over it;

c) first 12 octets of the result of step 5.b is the MAC.

The msgAuthenticationParameters field is replaced with the MAC value that was saved in step 2.

6) Then the newly calculated MAC is compared with the MAC saved in step 2. If they do not match, then an failure and an errorIndication (authenticationFailure) is returned to the calling module.

7) The authenticatedWholeMsg and statusInformation indicating success are then returned to the caller.

7. HMAC-SHA-96 Authentication Protocol

This section describes the HMAC-SHA-96 authentication protocol. This protocol uses the SHA hash-function which is described in [SHA-NIST], in HMAC mode described in [RFC2104], truncating the output to 96 bits.

This protocol is identified by usmHMACSHAAuthProtocol.

Over time, other authentication protocols may be defined either as a replacement of this protocol or in addition to this protocol.

7.1. Mechanisms

- In support of data integrity, a message digest algorithm is required. A digest is calculated over an appropriate portion of an SNMP message and included as part of the message sent to the recipient.

- In support of data origin authentication and data integrity, a secret value is prepended to the SNMP message prior to computing the digest; the calculated digest is then partially inserted into the message prior to transmission. The prepended secret is not transmitted. The secret value is shared by all SNMP engines authorized to originate messages on behalf of the appropriate user.

7.1.1. Digest Authentication Mechanism

The Digest Authentication Mechanism defined in this memo provides for:

- verification of the integrity of a received message, i.e., the message received is the message sent.

  The integrity of the message is protected by computing a digest over an appropriate portion of the message. The digest is computed by the originator of the message, transmitted with the message, and verified by the recipient of the message.

- verification of the user on whose behalf the message was generated.

  A secret value known only to SNMP engines authorized to generate messages on behalf of a user is used in HMAC mode (see [RFC2104]). It also recommends the hash-function output used as Message Authentication Code, to be truncated.

This mechanism uses the SHA [SHA-NIST] message digest algorithm. A 160-bit SHA digest is calculated in a special (HMAC) way over the designated portion of an SNMP message and the first 96 bits of this digest is included as part of the message sent to the recipient. The size of the digest carried in a message is 12 octets. The size of the private authentication key (the secret) is 20 octets. For the details see section 7.3.

7.2. Elements of the HMAC-SHA-96 Authentication Protocol

This section contains definitions required to realize the authentication module defined in this section of this memo.

7.2.1. Users

Authentication using this authentication protocol makes use of a defined set of userNames. For any user on whose behalf a message must be authenticated at a particular SNMP engine, that SNMP engine must have knowledge of that user. An SNMP engine that wishes to communicate with another SNMP engine must also have knowledge of a user known to that engine, including knowledge of the applicable attributes of that user.

A user and its attributes are defined as follows:

<userName>
  A string representing the name of the user.
<authKey>
  A user's secret key to be used when calculating a digest.
  It MUST be 20 octets long for SHA.

7.2.2. msgAuthoritativeEngineID

The msgAuthoritativeEngineID value contained in an authenticated message specifies the authoritative SNMP engine for that particular message (see the definition of SnmpEngineID in the SNMP Architecture document [RFC3411]).

The user's (private) authentication key is normally different at each authoritative SNMP engine and so the snmpEngineID is used to select the proper key for the authentication process.

7.2.3. SNMP Messages Using this Authentication Protocol

Messages using this authentication protocol carry a msgAuthenticationParameters field as part of the msgSecurityParameters. For this protocol, the msgAuthenticationParameters field is the serialized OCTET STRING representing the first 12 octets of HMAC-SHA-96 output done over the wholeMsg.

The digest is calculated over the wholeMsg so if a message is authenticated, that also means that all the fields in the message are intact and have not been tampered with.

7.2.4. Services Provided by the HMAC-SHA-96 Authentication Module

This section describes the inputs and outputs that the HMAC-SHA-96

Authentication module expects and produces when the User-based
Security module calls the HMAC-SHA-96 Authentication module for
services.

7.2.4.1. Services for Generating an Outgoing SNMP Message

HMAC-SHA-96 authentication protocol assumes that the selection of the
authKey is done by the caller and that the caller passes the secret
key to be used.

Upon completion the authentication module returns statusInformation
and, if the message digest was correctly calculated, the wholeMsg
with the digest inserted at the proper place. The abstract service
primitive is:

```
statusInformation =              -- success or failure
  authenticateOutgoingMsg(
  IN   authKey                   -- secret key for authentication
  IN   wholeMsg                  -- unauthenticated complete message
  OUT  authenticatedWholeMsg     -- complete authenticated message
       )
```

The abstract data elements are:

statusInformation
  An indication of whether the authentication process was successful.
  If not it is an indication of the problem.

authKey
  The secret key to be used by the authentication algorithm. The
  length of this key MUST be 20 octets.

wholeMsg
  The message to be authenticated.

authenticatedWholeMsg
  The authenticated message (including inserted digest) on output.

Note, that authParameters field is filled by the authentication
module and this field should be already present in the wholeMsg

before the Message Authentication Code (MAC) is generated.

7.2.4.2. Services for Processing an Incoming SNMP Message

HMAC-SHA-96 authentication protocol assumes that the selection of the authKey is done by the caller and that the caller passes the secret key to be used.

Upon completion the authentication module returns statusInformation and, if the message digest was correctly calculated, the wholeMsg as it was processed. The abstract service primitive is:

```
statusInformation =            -- success or failure
  authenticateIncomingMsg(
  IN   authKey                 -- secret key for authentication
  IN   authParameters          -- as received on the wire
  IN   wholeMsg                -- as received on the wire
  OUT  authenticatedWholeMsg   -- complete authenticated message
     )
```

The abstract data elements are:

statusInformation
  An indication of whether the authentication process was successful.
  If not it is an indication of the problem.

authKey
  The secret key to be used by the authentication algorithm. The
  length of this key MUST be 20 octets.

authParameters
  The authParameters from the incoming message.

wholeMsg
  The message to be authenticated on input and the authenticated
  message on output.

authenticatedWholeMsg
  The whole message after the authentication check is complete.

7.3. Elements of Procedure

This section describes the procedures for the HMAC-SHA-96 authentication protocol.

7.3.1. Processing an Outgoing Message

This section describes the procedure followed by an SNMP engine whenever it must authenticate an outgoing message using the usmHMACSHAAuthProtocol.

1) The msgAuthenticationParameters field is set to the serialization, according to the rules in [RFC3417], of an OCTET STRING containing 12 zero octets.

2) From the secret authKey, two keys K1 and K2 are derived:

   a) extend the authKey to 64 octets by appending 44 zero octets; save it as extendedAuthKey

   b) obtain IPAD by replicating the octet 0x36 64 times;

   c) obtain K1 by XORing extendedAuthKey with IPAD;

   d) obtain OPAD by replicating the octet 0x5C 64 times;

   e) obtain K2 by XORing extendedAuthKey with OPAD.

3) Prepend K1 to the wholeMsg and calculate the SHA digest over it according to [SHA-NIST].

4) Prepend K2 to the result of the step 4 and calculate SHA digest over it according to [SHA-NIST]. Take the first 12 octets of the final digest - this is Message Authentication Code (MAC).

5) Replace the msgAuthenticationParameters field with MAC obtained in the step 5.

6) The authenticatedWholeMsg is then returned to the caller together with statusInformation indicating success.

7.3.2. Processing an Incoming Message

This section describes the procedure followed by an SNMP engine whenever it must authenticate an incoming message using the usmHMACSHAAuthProtocol.

1) If the digest received in the msgAuthenticationParameters field is not 12 octets long, then an failure and an errorIndication (authenticationError) is returned to the calling module.

2) The MAC received in the msgAuthenticationParameters field is saved.

3) The digest in the msgAuthenticationParameters field is replaced by the 12 zero octets.

4) From the secret authKey, two keys K1 and K2 are derived:

   a) extend the authKey to 64 octets by appending 44 zero octets; save it as extendedAuthKey

   b) obtain IPAD by replicating the octet 0x36 64 times;

   c) obtain K1 by XORing extendedAuthKey with IPAD;

   d) obtain OPAD by replicating the octet 0x5C 64 times;

   e) obtain K2 by XORing extendedAuthKey with OPAD.

5) The MAC is calculated over the wholeMsg:

   a) prepend K1 to the wholeMsg and calculate the SHA digest over it;

   b) prepend K2 to the result of step 5.a and calculate the SHA digest over it;

   c) first 12 octets of the result of step 5.b is the MAC.

   The msgAuthenticationParameters field is replaced with the MAC value that was saved in step 2.

6) The the newly calculated MAC is compared with the MAC saved in step 2. If they do not match, then a failure and an errorIndication (authenticationFailure) are returned to the calling module.

7) The authenticatedWholeMsg and statusInformation indicating success are then returned to the caller.

8. CBC-DES Symmetric Encryption Protocol

This section describes the CBC-DES Symmetric Encryption Protocol. This protocol is the first privacy protocol defined for the User-based Security Model.

This protocol is identified by usmDESPrivProtocol.

Over time, other privacy protocols may be defined either as a replacement of this protocol or in addition to this protocol.

8.1. Mechanisms

- In support of data confidentiality, an encryption algorithm is required. An appropriate portion of the message is encrypted prior to being transmitted. The User-based Security Model specifies that the scopedPDU is the portion of the message that needs to be encrypted.

- A secret value in combination with a timeliness value is used to create the en/decryption key and the initialization vector. The secret value is shared by all SNMP engines authorized to originate messages on behalf of the appropriate user.

8.1.1. Symmetric Encryption Protocol

The Symmetric Encryption Protocol defined in this memo provides support for data confidentiality. The designated portion of an SNMP message is encrypted and included as part of the message sent to the recipient.

Two organizations have published specifications defining the DES:

the National Institute of Standards and Technology (NIST) [DES-NIST] and the American National Standards Institute [DES-ANSI]. There is a companion Modes of Operation specification for each definition ([DESO-NIST] and [DESO-ANSI], respectively).

The NIST has published three additional documents that implementors may find useful.

- There is a document with guidelines for implementing and using the DES, including functional specifications for the DES and its modes of operation [DESG-NIST].

- There is a specification of a validation test suite for the DES [DEST-NIST]. The suite is designed to test all aspects of the DES and is useful for pinpointing specific problems.

- There is a specification of a maintenance test for the DES [DESM-NIST]. The test utilizes a minimal amount of data and processing to test all components of the DES. It provides a simple yes-or-no indication of correct operation and is useful to run as part of an initialization step, e.g., when a computer re-boots.

8.1.1.1. DES key and Initialization Vector

The first 8 octets of the 16-octet secret (private privacy key) are used as a DES key. Since DES uses only 56 bits, the Least Significant Bit in each octet is disregarded.

The Initialization Vector for encryption is obtained using the following procedure.

The last 8 octets of the 16-octet secret (private privacy key) are used as pre-IV.

In order to ensure that the IV for two different packets encrypted by the same key, are not the same (i.e., the IV does not repeat) we need to "salt" the pre-IV with something unique per packet. An 8-octet string is used as the "salt". The concatenation of the generating SNMP engine's 32-bit snmpEngineBoots and a local 32-bit integer, that the encryption engine maintains, is input to the "salt". The 32-bit integer is initialized to an arbitrary value at boot time.

The 32-bit snmpEngineBoots is converted to the first 4 octets (Most Significant Byte first) of our "salt". The 32-bit integer is then converted to the last 4 octet (Most Significant Byte first) of our "salt". The resulting "salt" is then XOR-ed with the pre-IV to obtain the IV. The 8-octet "salt" is then put into the privParameters field encoded as an OCTET STRING. The "salt" integer is then modified. We recommend that it be incremented by one and wrap when it reaches the maximum value.

How exactly the value of the "salt" (and thus of the IV) varies, is an implementation issue, as long as the measures are taken to avoid producing a duplicate IV.

The "salt" must be placed in the privParameters field to enable the receiving entity to compute the correct IV and to decrypt the message.

8.1.1.2. Data Encryption

The data to be encrypted is treated as sequence of octets. Its length should be an integral multiple of 8 - and if it is not, the data is padded at the end as necessary. The actual pad value is irrelevant.

The data is encrypted in Cipher Block Chaining mode.

The plaintext is divided into 64-bit blocks.

The plaintext for each block is XOR-ed with the ciphertext of the previous block, the result is encrypted and the output of the encryption is the ciphertext for the block. This procedure is repeated until there are no more plaintext blocks.

For the very first block, the Initialization Vector is used instead of the ciphertext of the previous block.

8.1.1.3. Data Decryption

Before decryption, the encrypted data length is verified. If the length of the OCTET STRING to be decrypted is not an integral multiple of 8 octets, the decryption process is halted and an appropriate exception noted. When decrypting, the padding is ignored.

The first ciphertext block is decrypted, the decryption output is XOR-ed with the Initialization Vector, and the result is the first plaintext block.

For each subsequent block, the ciphertext block is decrypted, the decryption output is XOR-ed with the previous ciphertext block and the result is the plaintext block.

8.2. Elements of the DES Privacy Protocol

This section contains definitions required to realize the privacy module defined by this memo.

8.2.1. Users

Data en/decryption using this Symmetric Encryption Protocol makes use of a defined set of userNames. For any user on whose behalf a message must be en/decrypted at a particular SNMP engine, that SNMP engine must have knowledge of that user. An SNMP engine that wishes

to communicate with another SNMP engine must also have knowledge of a user known to that SNMP engine, including knowledge of the applicable attributes of that user.

A user and its attributes are defined as follows:

<userName>
An octet string representing the name of the user.

<privKey>
A user's secret key to be used as input for the DES key and IV. The length of this key MUST be 16 octets.

8.2.2. msgAuthoritativeEngineID

The msgAuthoritativeEngineID value contained in an authenticated message specifies the authoritative SNMP engine for that particular message (see the definition of SnmpEngineID in the SNMP Architecture document [RFC3411]).

The user's (private) privacy key is normally different at each authoritative SNMP engine and so the snmpEngineID is used to select the proper key for the en/decryption process.

8.2.3. SNMP Messages Using this Privacy Protocol

Messages using this privacy protocol carry a msgPrivacyParameters field as part of the msgSecurityParameters. For this protocol, the msgPrivacyParameters field is the serialized OCTET STRING representing the "salt" that was used to create the IV.

8.2.4. Services Provided by the DES Privacy Module

This section describes the inputs and outputs that the DES Privacy module expects and produces when the User-based Security module invokes the DES Privacy module for services.

8.2.4.1. Services for Encrypting Outgoing Data

This DES privacy protocol assumes that the selection of the privKey is done by the caller and that the caller passes the secret key to be used.

Upon completion the privacy module returns statusInformation and, if the encryption process was successful, the encryptedPDU and the msgPrivacyParameters encoded as an OCTET STRING. The abstract service primitive is:

```
statusInformation =              -- success of failure
  encryptData(
  IN    encryptKey               -- secret key for encryption
  IN    dataToEncrypt            -- data to encrypt (scopedPDU)
  OUT   encryptedData            -- encrypted data (encryptedPDU)
  OUT   privParameters           -- filled in by service provider
        )
```

The abstract data elements are:

statusInformation
  An indication of the success or failure of the encryption process.
  In case of failure, it is an indication of the error.

encryptKey
  The secret key to be used by the encryption algorithm. The length
  of this key MUST be 16 octets.

dataToEncrypt
  The data that must be encrypted.

encryptedData
  The encrypted data upon successful completion.

privParameters
  The privParameters encoded as an OCTET STRING.

8.2.4.2. Services for Decrypting Incoming Data

This DES privacy protocol assumes that the selection of the privKey is done by the caller and that the caller passes the secret key to be used.

Upon completion the privacy module returns statusInformation and, if the decryption process was successful, the scopedPDU in plain text. The abstract service primitive is:

```
statusInformation =
  decryptData(
  IN    decryptKey               -- secret key for decryption
  IN    privParameters           -- as received on the wire
  IN    encryptedData            -- encrypted data (encryptedPDU)
  OUT   decryptedData            -- decrypted data (scopedPDU)
        )
```

The abstract data elements are:

statusInformation
An indication whether the data was successfully decrypted and if not an indication of the error.

decryptKey
The secret key to be used by the decryption algorithm. The length of this key MUST be 16 octets.

privParameters
The "salt" to be used to calculate the IV.

encryptedData
The data to be decrypted.

decryptedData
The decrypted data.

8.3. Elements of Procedure.

This section describes the procedures for the DES privacy protocol.

8.3.1. Processing an Outgoing Message

This section describes the procedure followed by an SNMP engine whenever it must encrypt part of an outgoing message using the usmDESPrivProtocol.

1) The secret cryptKey is used to construct the DES encryption key, the "salt" and the DES pre-IV (from which the IV is computed as described in section 8.1.1.1).

2) The privParameters field is set to the serialization according to the rules in [RFC3417] of an OCTET STRING representing the "salt" string.

3) The scopedPDU is encrypted (as described in section 8.1.1.2) and the encrypted data is serialized according to the rules in [RFC3417] as an OCTET STRING.

4) The serialized OCTET STRING representing the encrypted scopedPDU together with the privParameters and statusInformation indicating success is returned to the calling module.

8.3.2. Processing an Incoming Message

This section describes the procedure followed by an SNMP engine whenever it must decrypt part of an incoming message using the usmDESPrivProtocol.

1) If the privParameters field is not an 8-octet OCTET STRING, then an error indication (decryptionError) is returned to the calling module.

2) The "salt" is extracted from the privParameters field.

3) The secret cryptKey and the "salt" are then used to construct the DES decryption key and pre-IV (from which the IV is computed as described in section 8.1.1.1).

4) The encryptedPDU is then decrypted (as described in section 8.1.1.3).

5) If the encryptedPDU cannot be decrypted, then an error indication (decryptionError) is returned to the calling module.

6) The decrypted scopedPDU and statusInformation indicating success are returned to the calling module.

9. Intellectual Property

The IETF 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 this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

10. Acknowledgements

This document is the result of the efforts of the SNMPv3 Working Group. Some special thanks are in order to the following SNMPv3 WG members:

Harald Tveit Alvestrand (Maxware)
Dave Battle (SNMP Research, Inc.)
Alan Beard (Disney Worldwide Services)
Paul Berrevoets (SWI Systemware/Halcyon Inc.)
Martin Bjorklund (Ericsson)
Uri Blumenthal (IBM T.J. Watson Research Center)
Jeff Case (SNMP Research, Inc.)
John Curran (BBN)
Mike Daniele (Compaq Computer Corporation))
T. Max Devlin (Eltrax Systems)
John Flick (Hewlett Packard)
Rob Frye (MCI)
Wes Hardaker (U.C.Davis, Information Technology - D.C.A.S.)
David Harrington (Cabletron Systems Inc.)
Lauren Heintz (BMC Software, Inc.)
N.C. Hien (IBM T.J. Watson Research Center)
Michael Kirkham (InterWorking Labs, Inc.)
Dave Levi (SNMP Research, Inc.)

Louis A Mamakos (UUNET Technologies Inc.)
Joe Marzot (Nortel Networks)
Paul Meyer (Secure Computing Corporation)
Keith McCloghrie (Cisco Systems)
Bob Moore (IBM)
Russ Mundy (TIS Labs at Network Associates)
Bob Natale (ACE*COMM Corporation)
Mike O'Dell (UUNET Technologies Inc.)
Dave Perkins (DeskTalk)
Peter Polkinghorne (Brunel University)
Randy Presuhn (BMC Software, Inc.)
David Reeder (TIS Labs at Network Associates)
David Reid (SNMP Research, Inc.)
Aleksey Romanov (Quality Quorum)
Shawn Routhier (Epilogue)
Juergen Schoenwaelder (TU Braunschweig)
Bob Stewart (Cisco Systems)
Mike Thatcher (Independent Consultant)
Bert Wijnen (IBM T.J. Watson Research Center)

The document is based on recommendations of the IETF Security and Administrative Framework Evolution for SNMP Advisory Team. Members of that Advisory Team were:

David Harrington (Cabletron Systems Inc.)
Jeff Johnson (Cisco Systems)
David Levi (SNMP Research Inc.)
John Linn (Openvision)
Russ Mundy (Trusted Information Systems) chair
Shawn Routhier (Epilogue)
Glenn Waters (Nortel)
Bert Wijnen (IBM T. J. Watson Research Center)

As recommended by the Advisory Team and the SNMPv3 Working Group Charter, the design incorporates as much as practical from previous RFCs and drafts. As a result, special thanks are due to the authors of previous designs known as SNMPv2u and SNMPv2*:

Jeff Case (SNMP Research, Inc.)
David Harrington (Cabletron Systems Inc.)
David Levi (SNMP Research, Inc.)
Keith McCloghrie (Cisco Systems)
Brian O'Keefe (Hewlett Packard)
Marshall T. Rose (Dover Beach Consulting)
Jon Saperia (BGS Systems Inc.)
Steve Waldbusser (International Network Services)
Glenn W. Waters (Bell-Northern Research Ltd.)

11. Security Considerations

11.1. Recommended Practices

This section describes practices that contribute to the secure, effective operation of the mechanisms defined in this memo.

- An SNMP engine must discard SNMP Response messages that do not correspond to any currently outstanding Request message. It is the responsibility of the Message Processing module to take care of this. For example it can use a msgID for that.

  An SNMP Command Generator Application must discard any Response Class PDU for which there is no currently outstanding Confirmed Class PDU; for example for SNMPv2 [RFC3416] PDUs, the request-id component in the PDU can be used to correlate Responses to outstanding Requests.

  Although it would be typical for an SNMP engine and an SNMP Command Generator Application to do this as a matter of course, when using these security protocols it is significant due to the possibility of message duplication (malicious or otherwise).

- If an SNMP engine uses a msgID for correlating Response messages to outstanding Request messages, then it MUST use different msgIDs in all such Request messages that it sends out during a Time Window

(150 seconds) period.

A Command Generator or Notification Originator Application MUST use different request-ids in all Request PDUs that it sends out during a TimeWindow (150 seconds) period.

This must be done to protect against the possibility of message duplication (malicious or otherwise).

For example, starting operations with a msgID and/or request-id value of zero is not a good idea. Initializing them with an unpredictable number (so they do not start out the same after each reboot) and then incrementing by one would be acceptable.

- An SNMP engine should perform time synchronization using authenticated messages in order to protect against the possibility of message duplication (malicious or otherwise).

- When sending state altering messages to a managed authoritative SNMP engine, a Command Generator Application should delay sending successive messages to that managed SNMP engine until a positive acknowledgement is received for the previous message or until the previous message expires.

No message ordering is imposed by the SNMP. Messages may be received in any order relative to their time of generation and each will be processed in the ordered received. Note that when an authenticated message is sent to a managed SNMP engine, it will be valid for a period of time of approximately 150 seconds under normal circumstances, and is subject to replay during this period. Indeed, an SNMP engine and SNMP Command Generator Applications must cope with the loss and re-ordering of messages resulting from anomalies in the network as a matter of course.

However, a managed object, snmpSetSerialNo [RFC3418], is specifically defined for use with SNMP Set operations in order to provide a mechanism to ensure that the processing of SNMP messages occurs in a specific order.

- The frequency with which the secrets of a User-based Security Model user should be changed is indirectly related to the frequency of their use.

  Protecting the secrets from disclosure is critical to the overall security of the protocols. Frequent use of a secret provides a continued source of data that may be useful to a cryptanalyst in exploiting known or perceived weaknesses in an algorithm. Frequent changes to the secret avoid this vulnerability.

  Changing a secret after each use is generally regarded as the most secure practice, but a significant amount of overhead may be associated with that approach.

  Note, too, in a local environment the threat of disclosure may be less significant, and as such the changing of secrets may be less frequent. However, when public data networks are used as the communication paths, more caution is prudent.

11.2 Defining Users

The mechanisms defined in this document employ the notion of users on whose behalf messages are sent. How "users" are defined is subject to the security policy of the network administration. For example, users could be individuals (e.g., "joe" or "jane"), or a particular role (e.g., "operator" or "administrator"), or a combination (e.g., "joe-operator", "jane-operator" or "joe-admin"). Furthermore, a user may be a logical entity, such as an SNMP Application or a set of SNMP Applications, acting on behalf of an individual or role, or set of individuals, or set of roles, including combinations.

Appendix A describes an algorithm for mapping a user "password" to a 16/20 octet value for use as either a user's authentication key or privacy key (or both). Note however, that using the same password (and therefore the same key) for both authentication and privacy is very poor security practice and should be strongly discouraged. Passwords are often generated, remembered, and input by a human. Human-generated passwords may be less than the 16/20 octets required by the authentication and privacy protocols, and brute force attacks can be quite easy on a relatively short ASCII character set. Therefore, the algorithm is Appendix A performs a transformation on the password. If the Appendix A algorithm is used, SNMP implementations (and SNMP configuration applications) must ensure that passwords are at least 8 characters in length. Please note that longer passwords with repetitive strings may result in exactly the same key. For example, a password 'bertbert' will result in exactly the same key as password 'bertbertbert'.

Because the Appendix A algorithm uses such passwords (nearly) directly, it is very important that they not be easily guessed. It is suggested that they be composed of mixed-case alphanumeric and punctuation characters that don't form words or phrases that might be found in a dictionary. Longer passwords improve the security of the system. Users may wish to input multiword phrases to make their password string longer while ensuring that it is memorable.

Since it is infeasible for human users to maintain different passwords for every SNMP engine, but security requirements strongly discourage having the same key for more than one SNMP engine, the User-based Security Model employs a compromise proposed in [Localized-key]. It derives the user keys for the SNMP engines from user's password in such a way that it is practically impossible to either determine the user's password, or user's key for another SNMP engine from any combination of user's keys on SNMP engines.

Note however, that if user's password is disclosed, then key localization will not help and network security may be compromised in this case. Therefore a user's password or non-localized key MUST NOT be stored on a managed device/node. Instead the localized key SHALL be stored (if at all), so that, in case a device does get compromised, no other managed or managing devices get compromised.

11.3. Conformance

To be termed a "Secure SNMP implementation" based on the User-based Security Model, an SNMP implementation MUST:

- implement one or more Authentication Protocol(s). The HMAC-MD5-96 and HMAC-SHA-96 Authentication Protocols defined in this memo are examples of such protocols.

- to the maximum extent possible, prohibit access to the secret(s) of each user about which it maintains information in a Local Configuration Datastore (LCD) under all circumstances except as required to generate and/or validate SNMP messages with respect to that user.

- implement the key-localization mechanism.

- implement the SNMP-USER-BASED-SM-MIB.

In addition, an authoritative SNMP engine SHOULD provide initial configuration in accordance with Appendix A.1.

Implementation of a Privacy Protocol (the DES Symmetric Encryption Protocol defined in this memo is one such protocol) is optional.

11.4. Use of Reports

The use of unsecure reports (i.e., sending them with a securityLevel of noAuthNoPriv) potentially exposes a non-authoritative SNMP engine to some form of attacks. Some people consider these denial of service attacks, others don't. An installation should evaluate the risk involved before deploying unsecure Report PDUs.

11.5 Access to the SNMP-USER-BASED-SM-MIB

The objects in this MIB may be considered sensitive in many environments. Specifically the objects in the usmUserTable contain information about users and their authentication and privacy protocols. It is important to closely control (both read and write) access to these MIB objects by using appropriately configured Access Control models (for example the View-based Access Control Model as specified in [RFC3415]).

12. References

12.1 Normative References

[RFC1321] Rivest, R., "Message Digest Algorithm MD5", RFC 1321, April 1992.

[RFC2104] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3411] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3412] Case, J., Harrington, D., Presuhn, R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, December 2002.

[RFC3415] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

[RFC3416] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

[RFC3417] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3417, December 2002.

[RFC3418] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[DES-NIST] Data Encryption Standard, National Institute of Standards and Technology. Federal Information Processing Standard (FIPS) Publication 46-1. Supersedes FIPS Publication 46, (January, 1977; reaffirmed January, 1988).

[DESO-NIST] DES Modes of Operation, National Institute of Standards and Technology. Federal Information Processing Standard (FIPS) Publication 81, (December, 1980).

[SHA-NIST] Secure Hash Algorithm. NIST FIPS 180-1, (April, 1995) http://csrc.nist.gov/fips/fip180-1.txt (ASCII) http://csrc.nist.gov/fips/fip180-1.ps (Postscript)

12.1 Informative References

[Localized-Key] U. Blumenthal, N. C. Hien, B. Wijnen "Key Derivation for Network Management Applications" IEEE Network Magazine, April/May issue, 1997.

[DES-ANSI] Data Encryption Algorithm, American National Standards Institute. ANSI X3.92-1981, (December, 1980).

[DESO-ANSI] Data Encryption Algorithm - Modes of Operation, American National Standards Institute. ANSI X3.106-1983, (May 1983).

[DESG-NIST] Guidelines for Implementing and Using the NBS Data Encryption Standard, National Institute of Standards

and Technology. Federal Information Processing
Standard (FIPS) Publication 74, (April, 1981).

[DEST-NIST] Validating the Correctness of Hardware
Implementations of the NBS Data Encryption Standard,
National Institute of Standards and Technology.
Special Publication 500-20.

[DESM-NIST] Maintenance Testing for the Data Encryption Standard,
National Institute of Standards and Technology.
Special Publication 500-61, (August, 1980).

[RFC3174] Eastlake, D. 3rd and P. Jones, "US Secure Hash
Algorithm 1 (SHA1)", RFC 3174, September 2001.

APPENDIX A - Installation

A.1. SNMP engine Installation Parameters

During installation, an authoritative SNMP engine SHOULD (in the meaning as defined in [RFC2119]) be configured with several initial parameters. These include:

1) A Security Posture

The choice of security posture determines if initial configuration is implemented and if so how. One of three possible choices is selected:

minimum-secure,
semi-secure,
very-secure (i.e., no-initial-configuration)

In the case of a very-secure posture, there is no initial configuration, and so the following steps are irrelevant.

2) One or More Secrets

These are the authentication/privacy secrets for the first user to be configured.

One way to accomplish this is to have the installer enter a "password" for each required secret. The password is then algorithmically converted into the required secret by:

- forming a string of length 1,048,576 octets by repeating the value of the password as often as necessary, truncating accordingly, and using the resulting string as the input to the MD5 algorithm [RFC1321]. The resulting digest, termed "digest1", is used in the next step.

- a second string is formed by concatenating digest1, the SNMP engine's snmpEngineID value, and digest1. This string is used as input to the MD5 algorithm [RFC1321].

  The resulting digest is the required secret (see Appendix A.2).

With these configured parameters, the SNMP engine instantiates the following usmUserEntry in the usmUserTable:

	no privacy support	privacy support
	------------------	---------------
usmUserEngineID	localEngineID	localEngineID
usmUserName	"initial"	"initial"
usmUserSecurityName	"initial"	"initial"
usmUserCloneFrom	ZeroDotZero	ZeroDotZero
usmUserAuthProtocol	usmHMACMD5AuthProtocol	usmHMACMD5AuthProtocol
usmUserAuthKeyChange	""	""
usmUserOwnAuthKeyChange	""	""
usmUserPrivProtocol	none	usmDESPrivProtocol
usmUserPrivKeyChange	""	""
usmUserOwnPrivKeyChange	""	""
usmUserPublic	""	""
usmUserStorageType	anyValidStorageType	anyValidStorageType
usmUserStatus	active	active

It is recommended to also instantiate a set of template usmUserEntries which can be used as clone-from users for newly created usmUserEntries. These are the two suggested entries:

	no privacy support	privacy support
	------------------	---------------
usmUserEngineID	localEngineID	localEngineID
usmUserName	"templateMD5"	"templateMD5"
usmUserSecurityName	"templateMD5"	"templateMD5"
usmUserCloneFrom	ZeroDotZero	ZeroDotZero
usmUserAuthProtocol	usmHMACMD5AuthProtocol	usmHMACMD5AuthProtocol
usmUserAuthKeyChange	""	""
usmUserOwnAuthKeyChange	""	""
usmUserPrivProtocol	none	usmDESPrivProtocol
usmUserPrivKeyChange	""	""
usmUserOwnPrivKeyChange	""	""
usmUserPublic	""	""
usmUserStorageType	permanent	permanent
usmUserStatus	active	active

```
                      no privacy support     privacy support
                      ------------------     ---------------
  usmUserEngineID        localEngineID         localEngineID
  usmUserName            "templateSHA"         "templateSHA"
  usmUserSecurityName    "templateSHA"         "templateSHA"
  usmUserCloneFrom       ZeroDotZero           ZeroDotZero
  usmUserAuthProtocol    usmHMACSHAAuthProtocol usmHMACSHAAuthProtocol
  usmUserAuthKeyChange   ""                    ""
  usmUserOwnAuthKeyChange ""                   ""
  usmUserPrivProtocol    none                  usmDESPrivProtocol
  usmUserPrivKeyChange   ""                    ""
  usmUserOwnPrivKeyChange ""                   ""
  usmUserPublic          ""                    ""
  usmUserStorageType     permanent             permanent
  usmUserStatus          active                active
```

A.2. Password to Key Algorithm

A sample code fragment (section A.2.1) demonstrates the password to key algorithm which can be used when mapping a password to an authentication or privacy key using MD5. The reference source code of MD5 is available in [RFC1321].

Another sample code fragment (section A.2.2) demonstrates the password to key algorithm which can be used when mapping a password to an authentication or privacy key using SHA (documented in SHA-NIST).

An example of the results of a correct implementation is provided (section A.3) which an implementor can use to check if his implementation produces the same result.

Blumenthal & Wijnen   Standards Track   [Page 80]

RFC 3414   USM for SNMPv3   December 2002

A.2.1. Password to Key Sample Code for MD5

```
 void password_to_key_md5(
    u_char *password,    /* IN */
    u_int   passwordlen, /* IN */
    u_char *engineID,    /* IN  - pointer to snmpEngineID  */
    u_int   engineLength,/* IN  - length of snmpEngineID */
    u_char *key)         /* OUT - pointer to caller 16-octet buffer */
 {
    MD5_CTX     MD;
    u_char     *cp, password_buf[64];
    u_long      password_index = 0;
    u_long      count = 0, i;

    MD5Init (&MD);   /* initialize MD5 */

    /**********************************************/
    /* Use while loop until we've done 1 Megabyte */
    /**********************************************/
    while (count < 1048576) {
       cp = password_buf;
       for (i = 0; i < 64; i++) {
           /*************************************************/
           /* Take the next octet of the password, wrapping */
           /* to the beginning of the password as necessary.*/
           /*************************************************/
           *cp++ = password[password_index++ % passwordlen];
       }
       MD5Update (&MD, password_buf, 64);
       count += 64;
    }
    MD5Final (key, &MD);          /* tell MD5 we're done */
```

```
    /*****************************************************/
    /* Now localize the key with the engineID and pass   */
    /* through MD5 to produce final key                  */
    /* May want to ensure that engineLength <= 32,       */
    /* otherwise need to use a buffer larger than 64     */
    /*****************************************************/
    memcpy(password_buf, key, 16);
    memcpy(password_buf+16, engineID, engineLength);
    memcpy(password_buf+16+engineLength, key, 16);

    MD5Init(&MD);
    MD5Update(&MD, password_buf, 32+engineLength);
    MD5Final(key, &MD);
    return;
  }
```

Blumenthal & Wijnen Standards Track [Page 81]

RFC 3414 USM for SNMPv3 December 2002

A.2.2. Password to Key Sample Code for SHA

```
  void password_to_key_sha(
    u_char *password,    /* IN */
    u_int   passwordlen, /* IN */
    u_char *engineID,    /* IN  - pointer to snmpEngineID  */
    u_int   engineLength,/* IN  - length of snmpEngineID */
    u_char *key)         /* OUT - pointer to caller 20-octet buffer */
  {
    SHA_CTX     SH;
    u_char     *cp, password_buf[72];
    u_long      password_index = 0;
    u_long      count = 0, i;

    SHAInit (&SH);   /* initialize SHA */

    /**********************************************/
    /* Use while loop until we've done 1 Megabyte */
    /**********************************************/
    while (count < 1048576) {
      cp = password_buf;
      for (i = 0; i < 64; i++) {
          /*************************************************/
          /* Take the next octet of the password, wrapping */
```

```
        /* to the beginning of the password as necessary.*/
        /*************************************************/
        *cp++ = password[password_index++ % passwordlen];
      }
      SHAUpdate (&SH, password_buf, 64);
      count += 64;
    }
    SHAFinal (key, &SH);          /* tell SHA we're done */

    /*****************************************************/
    /* Now localize the key with the engineID and pass   */
    /* through SHA to produce final key                  */
    /* May want to ensure that engineLength <= 32,       */
    /* otherwise need to use a buffer larger than 72     */
    /*****************************************************/
    memcpy(password_buf, key, 20);
    memcpy(password_buf+20, engineID, engineLength);
    memcpy(password_buf+20+engineLength, key, 20);

    SHAInit(&SH);
    SHAUpdate(&SH, password_buf, 40+engineLength);
    SHAFinal(key, &SH);
    return;
  }
```


A.3. Password to Key Sample Results

A.3.1. Password to Key Sample Results using MD5

The following shows a sample output of the password to key algorithm for a 16-octet key using MD5.

With a password of "maplesyrup" the output of the password to key algorithm before the key is localized with the SNMP engine's snmpEngineID is:

  '9f af 32 83 88 4e 92 83 4e bc 98 47 d8 ed d9 63'H

After the intermediate key (shown above) is localized with the snmpEngineID value of:

'00 00 00 00 00 00 00 00 00 00 00 02'H

the final output of the password to key algorithm is:

'52 6f 5e ed 9f cc e2 6f 89 64 c2 93 07 87 d8 2b'H

A.3.2. Password to Key Sample Results using SHA

The following shows a sample output of the password to key algorithm for a 20-octet key using SHA.

With a password of "maplesyrup" the output of the password to key algorithm before the key is localized with the SNMP engine's snmpEngineID is:

'9f b5 cc 03 81 49 7b 37 93 52 89 39 ff 78 8d 5d 79 14 52 11'H

After the intermediate key (shown above) is localized with the snmpEngineID value of:

'00 00 00 00 00 00 00 00 00 00 00 02'H

the final output of the password to key algorithm is:

'66 95 fe bc 92 88 e3 62 82 23 5f c7 15 1f 12 84 97 b3 8f 3f'H

A.4. Sample Encoding of msgSecurityParameters

The msgSecurityParameters in an SNMP message are represented as an OCTET STRING. This OCTET STRING should be considered opaque outside a specific Security Model.

The User-based Security Model defines the contents of the OCTET STRING as a SEQUENCE (see section 2.4).

Given these two properties, the following is an example of they msgSecurityParameters for the User-based Security Model, encoded as an OCTET STRING:

04 <length>

```
   30 <length>
   04 <length> <msgAuthoritativeEngineID>
   02 <length> <msgAuthoritativeEngineBoots>
   02 <length> <msgAuthoritativeEngineTime>
   04 <length> <msgUserName>
   04 0c       <HMAC-MD5-96-digest>
   04 08       <salt>
```

Here is the example once more, but now with real values (except for the digest in msgAuthenticationParameters and the salt in msgPrivacyParameters, which depend on variable data that we have not defined here):

```
   Hex Data                         Description
   --------------  -----------------------------------------------
   04 39           OCTET STRING,                  length 57
   30 37           SEQUENCE,                      length 55
   04 0c 80000002  msgAuthoritativeEngineID:      IBM
         01                                       IPv4 address
         09840301                                 9.132.3.1
   02 01 01        msgAuthoritativeEngineBoots:   1
   02 02 0101      msgAuthoritativeEngineTime:    257
   04 04 62657274  msgUserName:                   bert
   04 0c 01234567  msgAuthenticationParameters:   sample value
         89abcdef
         fedcba98
   04 08 01234567  msgPrivacyParameters:          sample value
         89abcdef
```

A.5. Sample keyChange Results

A.5.1. Sample keyChange Results using MD5

Let us assume that a user has a current password of "maplesyrup" as in section A.3.1. and let us also assume the snmpEngineID of 12 octets:

```
   '00 00 00 00 00 00 00 00 00 00 00 02'H
```

If we now want to change the password to "newsyrup", then we first calculate the key for the new password. It is as follows:

'01 ad d2 73 10 7c 4e 59 6b 4b 00 f8 2b 1d 42 a7'H

If we localize it for the above snmpEngineID, then the localized new key becomes:

'87 02 1d 7b d9 d1 01 ba 05 ea 6e 3b f9 d9 bd 4a'H

If we then use a (not so good, but easy to test) random value of:

'00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00'H

Then the value we must send for keyChange is:

'00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 88 05 61 51 41 67 6c c9 19 61 74 e7 42 a3 25 51'H

If this were for the privacy key, then it would be exactly the same.

A.5.2. Sample keyChange Results using SHA

Let us assume that a user has a current password of "maplesyrup" as in section A.3.2. and let us also assume the snmpEngineID of 12 octets:

'00 00 00 00 00 00 00 00 00 00 00 02'H

If we now want to change the password to "newsyrup", then we first calculate the key for the new password. It is as follows:

'3a 51 a6 d7 36 aa 34 7b 83 dc 4a 87 e3 e5 5e e4 d6 98 ac 71'H

If we localize it for the above snmpEngineID, then the localized new key becomes:

'78 e2 dc ce 79 d5 94 03 b5 8c 1b ba a5 bf f4 63 91 f1 cd 25'H

If we then use a (not so good, but easy to test) random value of:

'00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00'H

Then the value we must send for keyChange is:

'00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 9c 10 17 f4 fd 48 3d 2d e8 d5 fa db f8 43 92 cb 06 45 70 51'

For the key used for privacy, the new nonlocalized key would be:

'3a 51 a6 d7 36 aa 34 7b 83 dc 4a 87 e3 e5 5e e4 d6 98 ac 71'H

For the key used for privacy, the new localized key would be (note that they localized key gets truncated to 16 octets for DES):

'78 e2 dc ce 79 d5 94 03 b5 8c 1b ba a5 bf f4 63'H

If we then use a (not so good, but easy to test) random value of:

'00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00'H

Then the value we must send for keyChange for the privacy key is:

'00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
'7e f8 d8 a4 c9 cd b2 6b 47 59 1c d8 52 ff 88 b5'H

B. Change Log

Changes made since RFC2574:

- Updated references
- Updated contact info
- Clarifications
  - to first constraint item 1) on page 6.
  - to usmUserCloneFrom DESCRIPTION clause
  - to securityName in section 2.1
- Fixed "command responder" into "command generator" in last para of DESCRIPTION clause of usmUserTable.

Changes made since RFC2274:

- Fixed msgUserName to allow size of zero and explain that this can be used for snmpEngineID discovery.
- Clarified section 3.1 steps 4.b, 5, 6 and 8.b.
- Clarified section 3.2 paragraph 2.
- Clarified section 3.2 step 7.a last paragraph, step 7.b.1 second bullet and step 7.b.2 third bullet.
- Clarified section 4 to indicate that discovery can use a userName

of zero length in unAuthenticated messages, whereas a valid
userName must be used in authenticated messages.

- Added REVISION clauses to MODULE-IDENTITY
- Clarified KeyChange TC by adding a note that localized keys must be used when calculating a KeyChange value.
- Added clarifying text to the DESCRIPTION clause of usmUserTable. Added text describes a recommended procedure for adding a new user.
- Clarified the use of usmUserCloneFrom object.

- Clarified how and under which conditions the usmUserAuthProtocol and usmUserPrivProtocol can be initialized and/or changed.
- Added comment on typical sizes for usmUserAuthKeyChange and usmUserPrivKeyChange. Also for usmUserOwnAuthKeyChange and usmUserOwnPrivKeyChange.
- Added clarifications to the DESCRIPTION clauses of usmUserAuthKeyChange, usmUserOwnAuthKeychange, usmUserPrivKeyChange and usmUserOwnPrivKeychange.
- Added clarification to DESCRIPTION clause of usmUserStorageType.
- Added clarification to DESCRIPTION clause of usmUserStatus.
- Clarified IV generation procedure in section 8.1.1.1 and in addition clarified section 8.3.1 step 1 and section 8.3.2. step 3.
- Clarified section 11.2 and added a warning that different size passwords with repetitive strings may result in same key.
- Added template users to appendix A for cloning process.
- Fixed C-code examples in Appendix A.
- Fixed examples of generated keys in Appendix A.
- Added examples of KeyChange values to Appendix A.
- Used PDU Classes instead of RFC1905 PDU types.
- Added text in the security section about Reports and Access Control to the MIB.
- Removed a incorrect note at the end of section 3.2 step 7.
- Added a note in section 3.2 step 3.
- Corrected various spelling errors and typos.
- Corrected procedure for 3.2 step 2.a)
- various clarifications.
- Fixed references to new/revised documents
- Change to no longer cache data that is not used

Editors' Addresses

Uri Blumenthal

Lucent Technologies
67 Whippany Rd.
Whippany, NJ 07981
USA

Phone: +1-973-386-2163
EMail: uri@lucent.com

Bert Wijnen
Lucent Technologies
Schagen 33
3461 GL Linschoten
Netherlands

Phone: +31-348-480-685
EMail: bwijnen@lucent.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the
Internet Society.

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/ref/rfc/v3/rfc3414.txt
No license file was found, but licenses were detected in source scan.

```
SNMPv2-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
   TimeTicks, Counter32, snmpModules, mib-2
       FROM SNMPv2-SMI
   DisplayString, TestAndIncr, TimeStamp

       FROM SNMPv2-TC
   MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
       FROM SNMPv2-CONF;

snmpMIB MODULE-IDENTITY
   LAST-UPDATED "200210160000Z"
   ORGANIZATION "IETF SNMPv3 Working Group"
```

```
CONTACT-INFO
       "WG-EMail:   snmpv3@lists.tislabs.com
        Subscribe:  snmpv3-request@lists.tislabs.com

        Co-Chair:   Russ Mundy
                    Network Associates Laboratories
        postal:     15204 Omega Drive, Suite 300
                    Rockville, MD 20850-4601
                    USA
        EMail:      mundy@tislabs.com
        phone:      +1 301 947-7107

        Co-Chair:   David Harrington
                    Enterasys Networks
        postal:     35 Industrial Way
                    P. O. Box 5005
                    Rochester, NH 03866-5005
                    USA
        EMail:      dbh@enterasys.com
        phone:      +1 603 337-2614

        Editor:     Randy Presuhn
                    BMC Software, Inc.
        postal:     2141 North First Street
                    San Jose, CA 95131
                    USA
        EMail:      randy_presuhn@bmc.com
        phone:      +1 408 546-1006"
DESCRIPTION
       "The MIB module for SNMP entities.

        Copyright (C) The Internet Society (2002). This
        version of this MIB module is part of RFC 3418;
        see the RFC itself for full legal notices.
       "
REVISION      "200210160000Z"
DESCRIPTION
       "This revision of this MIB module was published as
        RFC 3418."
REVISION      "199511090000Z"
DESCRIPTION
       "This revision of this MIB module was published as
        RFC 1907."
REVISION      "199304010000Z"
DESCRIPTION
       "The initial revision of this MIB module was published
       as RFC 1450."
::= { snmpModules 1 }
```

```
snmpMIBObjects OBJECT IDENTIFIER ::= { snmpMIB 1 }

--  ::= { snmpMIBObjects 1 }        this OID is obsolete
--  ::= { snmpMIBObjects 2 }        this OID is obsolete
--  ::= { snmpMIBObjects 3 }        this OID is obsolete

-- the System group
--
-- a collection of objects common to all managed systems.

system   OBJECT IDENTIFIER ::= { mib-2 1 }

sysDescr OBJECT-TYPE
   SYNTAX      DisplayString (SIZE (0..255))
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
           "A textual description of the entity.  This value should
           include the full name and version identification of
           the system's hardware type, software operating-system,
           and networking software."
   ::= { system 1 }

sysObjectID OBJECT-TYPE
   SYNTAX      OBJECT IDENTIFIER
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
           "The vendor's authoritative identification of the
           network management subsystem contained in the entity.
           This value is allocated within the SMI enterprises
           subtree (1.3.6.1.4.1) and provides an easy and
           unambiguous means for determining `what kind of box' is
           being managed.  For example, if vendor `Flintstones,
           Inc.' was assigned the subtree 1.3.6.1.4.1.424242,
           it could assign the identifier 1.3.6.1.4.1.424242.1.1
           to its `Fred Router'."
   ::= { system 2 }

sysUpTime OBJECT-TYPE
   SYNTAX      TimeTicks
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
           "The time (in hundredths of a second) since the
           network management portion of the system was last
           re-initialized."
```

```
  ::= { system 3 }

sysContact OBJECT-TYPE
  SYNTAX      DisplayString (SIZE (0..255))
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
          "The textual identification of the contact person for
          this managed node, together with information on how
          to contact this person.  If no contact information is
          known, the value is the zero-length string."
  ::= { system 4 }

sysName OBJECT-TYPE
  SYNTAX      DisplayString (SIZE (0..255))
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
          "An administratively-assigned name for this managed
          node.  By convention, this is the node's fully-qualified
          domain name.  If the name is unknown, the value is
          the zero-length string."
  ::= { system 5 }

sysLocation OBJECT-TYPE
  SYNTAX      DisplayString (SIZE (0..255))
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
          "The physical location of this node (e.g., 'telephone
          closet, 3rd floor').  If the location is unknown, the
          value is the zero-length string."
  ::= { system 6 }

sysServices OBJECT-TYPE
  SYNTAX      INTEGER (0..127)
  MAX-ACCESS  read-only
  STATUS      current
  DESCRIPTION
          "A value which indicates the set of services that this
          entity may potentially offer.  The value is a sum.

          This sum initially takes the value zero. Then, for
          each layer, L, in the range 1 through 7, that this node
          performs transactions for, 2 raised to (L - 1) is added
          to the sum.  For example, a node which performs only
          routing functions would have a value of 4 (2^(3-1)).
          In contrast, a node which is a host offering application
```

```
            services would have a value of 72 (2^(4-1) + 2^(7-1)).
            Note that in the context of the Internet suite of
            protocols, values should be calculated accordingly:

                 layer      functionality
                   1        physical (e.g., repeaters)
                   2        datalink/subnetwork (e.g., bridges)
                   3        internet (e.g., supports the IP)
                   4        end-to-end  (e.g., supports the TCP)
                   7        applications (e.g., supports the SMTP)

            For systems including OSI protocols, layers 5 and 6
            may also be counted."
    ::= { system 7 }

-- object resource information
--
-- a collection of objects which describe the SNMP entity's
-- (statically and dynamically configurable) support of
-- various MIB modules.

sysORLastChange OBJECT-TYPE
    SYNTAX     TimeStamp
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
            "The value of sysUpTime at the time of the most recent
            change in state or value of any instance of sysORID."
    ::= { system 8 }

sysORTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF SysOREntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
            "The (conceptual) table listing the capabilities of
            the local SNMP application acting as a command
            responder with respect to various MIB modules.
            SNMP entities having dynamically-configurable support
            of MIB modules will have a dynamically-varying number
            of conceptual rows."
    ::= { system 9 }

sysOREntry OBJECT-TYPE
    SYNTAX     SysOREntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
```

```
        "An entry (conceptual row) in the sysORTable."
    INDEX      { sysORIndex }
    ::= { sysORTable 1 }

SysOREntry ::= SEQUENCE {
    sysORIndex     INTEGER,
    sysORID        OBJECT IDENTIFIER,
    sysORDescr     DisplayString,
    sysORUpTime    TimeStamp
}

sysORIndex OBJECT-TYPE
    SYNTAX     INTEGER (1..2147483647)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The auxiliary variable used for identifying instances
        of the columnar objects in the sysORTable."
    ::= { sysOREntry 1 }

sysORID OBJECT-TYPE
    SYNTAX     OBJECT IDENTIFIER
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "An authoritative identification of a capabilities
        statement with respect to various MIB modules supported
        by the local SNMP application acting as a command
        responder."
    ::= { sysOREntry 2 }

sysORDescr OBJECT-TYPE
    SYNTAX     DisplayString
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "A textual description of the capabilities identified
        by the corresponding instance of sysORID."
    ::= { sysOREntry 3 }

sysORUpTime OBJECT-TYPE
    SYNTAX     TimeStamp
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The value of sysUpTime at the time this conceptual
        row was last instantiated."
    ::= { sysOREntry 4 }
```

```
-- the SNMP group
--
-- a collection of objects providing basic instrumentation and
-- control of an SNMP entity.

snmp     OBJECT IDENTIFIER ::= { mib-2 11 }

snmpInPkts OBJECT-TYPE
   SYNTAX     Counter32
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
           "The total number of messages delivered to the SNMP
           entity from the transport service."
   ::= { snmp 1 }

snmpInBadVersions OBJECT-TYPE
   SYNTAX     Counter32
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
           "The total number of SNMP messages which were delivered
           to the SNMP entity and were for an unsupported SNMP
           version."
   ::= { snmp 3 }

snmpInBadCommunityNames OBJECT-TYPE
   SYNTAX     Counter32
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
          "The total number of community-based SNMP messages (for
          example,  SNMPv1) delivered to the SNMP entity which
          used an SNMP community name not known to said entity.
          Also, implementations which authenticate community-based
          SNMP messages using check(s) in addition to matching
          the community name (for example, by also checking
          whether the message originated from a transport address
          allowed to use a specified community name) MAY include
          in this value the number of messages which failed the
          additional check(s).  It is strongly RECOMMENDED that

          the documentation for any security model which is used
          to authenticate community-based SNMP messages specify
          the precise conditions that contribute to this value."
   ::= { snmp 4 }
```

```
snmpInBadCommunityUses OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS read-only
  STATUS     current
  DESCRIPTION
       "The total number of community-based SNMP messages (for
       example, SNMPv1) delivered to the SNMP entity which
       represented an SNMP operation that was not allowed for
       the SNMP community named in the message.  The precise
       conditions under which this counter is incremented
       (if at all) depend on how the SNMP entity implements
       its access control mechanism and how its applications
       interact with that access control mechanism.  It is
       strongly RECOMMENDED that the documentation for any
       access control mechanism which is used to control access
       to and visibility of MIB instrumentation specify the
       precise conditions that contribute to this value."
  ::= { snmp 5 }

snmpInASNParseErrs OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS read-only
  STATUS     current
  DESCRIPTION
        "The total number of ASN.1 or BER errors encountered by
        the SNMP entity when decoding received SNMP messages."
  ::= { snmp 6 }

snmpEnableAuthenTraps OBJECT-TYPE
  SYNTAX      INTEGER { enabled(1), disabled(2) }
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION
        "Indicates whether the SNMP entity is permitted to
        generate authenticationFailure traps.  The value of this
        object overrides any configuration information; as such,
        it provides a means whereby all authenticationFailure
        traps may be disabled.

        Note that it is strongly recommended that this object
        be stored in non-volatile memory so that it remains
        constant across re-initializations of the network
        management system."
  ::= { snmp 30 }

snmpSilentDrops OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS read-only
```

```
  STATUS     current
  DESCRIPTION
        "The total number of Confirmed Class PDUs (such as
        GetRequest-PDUs, GetNextRequest-PDUs,
        GetBulkRequest-PDUs, SetRequest-PDUs, and
        InformRequest-PDUs) delivered to the SNMP entity which
        were silently dropped because the size of a reply
        containing an alternate Response Class PDU (such as a
        Response-PDU) with an empty variable-bindings field
        was greater than either a local constraint or the
        maximum message size associated with the originator of
        the request."
  ::= { snmp 31 }

snmpProxyDrops OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS read-only
  STATUS     current
  DESCRIPTION
         "The total number of Confirmed Class PDUs
         (such as GetRequest-PDUs, GetNextRequest-PDUs,
         GetBulkRequest-PDUs, SetRequest-PDUs, and
         InformRequest-PDUs) delivered to the SNMP entity which
         were silently dropped because the transmission of
         the (possibly translated) message to a proxy target
         failed in a manner (other than a time-out) such that
         no Response Class PDU (such as a Response-PDU) could
         be returned."
  ::= { snmp 32 }

-- information for notifications
--
-- a collection of objects which allow the SNMP entity, when
-- supporting a notification originator application,
-- to be configured to generate SNMPv2-Trap-PDUs.

snmpTrap       OBJECT IDENTIFIER ::= { snmpMIBObjects 4 }

snmpTrapOID OBJECT-TYPE
  SYNTAX     OBJECT IDENTIFIER
  MAX-ACCESS accessible-for-notify
  STATUS     current
  DESCRIPTION
         "The authoritative identification of the notification
         currently being sent.  This variable occurs as
         the second varbind in every SNMPv2-Trap-PDU and
         InformRequest-PDU."
  ::= { snmpTrap 1 }
```

```
--  ::= { snmpTrap 2 }   this OID is obsolete

snmpTrapEnterprise OBJECT-TYPE
   SYNTAX     OBJECT IDENTIFIER
   MAX-ACCESS accessible-for-notify
   STATUS     current
   DESCRIPTION
           "The authoritative identification of the enterprise
           associated with the trap currently being sent.  When an
           SNMP proxy agent is mapping an RFC1157 Trap-PDU
           into a SNMPv2-Trap-PDU, this variable occurs as the
           last varbind."
   ::= { snmpTrap 3 }

--  ::= { snmpTrap 4 }   this OID is obsolete

-- well-known traps

snmpTraps      OBJECT IDENTIFIER ::= { snmpMIBObjects 5 }

coldStart NOTIFICATION-TYPE
   STATUS  current
   DESCRIPTION
           "A coldStart trap signifies that the SNMP entity,
           supporting a notification originator application, is
           reinitializing itself and that its configuration may
           have been altered."
   ::= { snmpTraps 1 }

warmStart NOTIFICATION-TYPE
   STATUS  current
   DESCRIPTION
           "A warmStart trap signifies that the SNMP entity,
           supporting a notification originator application,
           is reinitializing itself such that its configuration
           is unaltered."
   ::= { snmpTraps 2 }

-- Note the linkDown NOTIFICATION-TYPE ::= { snmpTraps 3 }
-- and the linkUp NOTIFICATION-TYPE ::= { snmpTraps 4 }
-- are defined in RFC 2863 [RFC2863]

authenticationFailure NOTIFICATION-TYPE
   STATUS  current
   DESCRIPTION
           "An authenticationFailure trap signifies that the SNMP
            entity has received a protocol message that is not
```

```
            properly authenticated.  While all implementations
            of SNMP entities MAY be capable of generating this
            trap, the snmpEnableAuthenTraps object indicates
            whether this trap will be generated."
    ::= { snmpTraps 5 }

-- Note the egpNeighborLoss notification is defined
-- as { snmpTraps 6 } in RFC 1213

-- the set group
--
-- a collection of objects which allow several cooperating
-- command generator applications to coordinate their use of the
-- set operation.

snmpSet        OBJECT IDENTIFIER ::= { snmpMIBObjects 6 }

snmpSetSerialNo OBJECT-TYPE
    SYNTAX     TestAndIncr
    MAX-ACCESS read-write
    STATUS     current
    DESCRIPTION
           "An advisory lock used to allow several cooperating
           command generator applications to coordinate their
           use of the SNMP set operation.

           This object is used for coarse-grain coordination.
           To achieve fine-grain coordination, one or more similar
           objects might be defined within each MIB group, as
           appropriate."
    ::= { snmpSet 1 }

-- conformance information

snmpMIBConformance
              OBJECT IDENTIFIER ::= { snmpMIB 2 }

snmpMIBCompliances
              OBJECT IDENTIFIER ::= { snmpMIBConformance 1 }
snmpMIBGroups  OBJECT IDENTIFIER ::= { snmpMIBConformance 2 }

-- compliance statements

--    ::= { snmpMIBCompliances 1 }      this OID is obsolete
snmpBasicCompliance MODULE-COMPLIANCE
    STATUS  deprecated
    DESCRIPTION
           "The compliance statement for SNMPv2 entities which
```

```
        implement the SNMPv2 MIB.

        This compliance statement is replaced by
        snmpBasicComplianceRev2."
  MODULE  -- this module
    MANDATORY-GROUPS { snmpGroup, snmpSetGroup, systemGroup,
                       snmpBasicNotificationsGroup }

    GROUP   snmpCommunityGroup
    DESCRIPTION
        "This group is mandatory for SNMPv2 entities which
        support community-based authentication."
  ::= { snmpMIBCompliances 2 }

snmpBasicComplianceRev2 MODULE-COMPLIANCE
  STATUS  current
  DESCRIPTION
        "The compliance statement for SNMP entities which
        implement this MIB module."
  MODULE  -- this module
    MANDATORY-GROUPS { snmpGroup, snmpSetGroup, systemGroup,
                       snmpBasicNotificationsGroup }

    GROUP   snmpCommunityGroup
    DESCRIPTION
        "This group is mandatory for SNMP entities which
        support community-based authentication."

    GROUP   snmpWarmStartNotificationGroup
    DESCRIPTION
        "This group is mandatory for an SNMP entity which
        supports command responder applications, and is
        able to reinitialize itself such that its
        configuration is unaltered."
  ::= { snmpMIBCompliances 3 }

-- units of conformance

--  ::= { snmpMIBGroups 1 }        this OID is obsolete
--  ::= { snmpMIBGroups 2 }        this OID is obsolete
--  ::= { snmpMIBGroups 3 }        this OID is obsolete

--  ::= { snmpMIBGroups 4 }        this OID is obsolete

snmpGroup OBJECT-GROUP
  OBJECTS { snmpInPkts,
         snmpInBadVersions,
         snmpInASNParseErrs,
```

```
           snmpSilentDrops,
           snmpProxyDrops,
           snmpEnableAuthenTraps }
   STATUS  current
   DESCRIPTION
          "A collection of objects providing basic instrumentation
          and control of an SNMP entity."
   ::= { snmpMIBGroups 8 }

snmpCommunityGroup OBJECT-GROUP
   OBJECTS { snmpInBadCommunityNames,
           snmpInBadCommunityUses }
   STATUS  current
   DESCRIPTION
          "A collection of objects providing basic instrumentation
          of a SNMP entity which supports community-based
          authentication."
   ::= { snmpMIBGroups 9 }

snmpSetGroup OBJECT-GROUP
   OBJECTS { snmpSetSerialNo }
   STATUS  current
   DESCRIPTION
          "A collection of objects which allow several cooperating
          command generator applications to coordinate their
          use of the set operation."
   ::= { snmpMIBGroups 5 }

systemGroup OBJECT-GROUP
   OBJECTS { sysDescr, sysObjectID, sysUpTime,
           sysContact, sysName, sysLocation,
           sysServices,
           sysORLastChange, sysORID,
           sysORUpTime, sysORDescr }
   STATUS  current
   DESCRIPTION
          "The system group defines objects which are common to all
          managed systems."
   ::= { snmpMIBGroups 6 }

snmpBasicNotificationsGroup NOTIFICATION-GROUP
   NOTIFICATIONS { coldStart, authenticationFailure }
   STATUS      current
   DESCRIPTION
      "The basic notifications implemented by an SNMP entity
       supporting command responder applications."
   ::= { snmpMIBGroups 7 }
```

```
snmpWarmStartNotificationGroup NOTIFICATION-GROUP
  NOTIFICATIONS { warmStart }
  STATUS       current
  DESCRIPTION
    "An additional notification for an SNMP entity supporting
    command responder applications, if it is able to reinitialize
    itself such that its configuration is unaltered."
 ::= { snmpMIBGroups 11 }

snmpNotificationGroup OBJECT-GROUP
   OBJECTS { snmpTrapOID, snmpTrapEnterprise }
   STATUS  current
   DESCRIPTION
           "These objects are required for entities
           which support notification originator applications."
   ::= { snmpMIBGroups 12 }

-- definitions in RFC 1213 made obsolete by the inclusion of a
-- subset of the snmp group in this MIB

snmpOutPkts OBJECT-TYPE
   SYNTAX      Counter32
   MAX-ACCESS  read-only
   STATUS      obsolete
   DESCRIPTION
           "The total number of SNMP Messages which were
           passed from the SNMP protocol entity to the
           transport service."
   ::= { snmp 2 }

-- { snmp 7 } is not used

snmpInTooBigs OBJECT-TYPE
   SYNTAX      Counter32
   MAX-ACCESS  read-only
   STATUS      obsolete
   DESCRIPTION
           "The total number of SNMP PDUs which were
           delivered to the SNMP protocol entity and for
           which the value of the error-status field was
           `tooBig'."
   ::= { snmp 8 }

snmpInNoSuchNames OBJECT-TYPE
   SYNTAX      Counter32
   MAX-ACCESS  read-only
   STATUS      obsolete
   DESCRIPTION
```

```
        "The total number of SNMP PDUs which were
        delivered to the SNMP protocol entity and for
        which the value of the error-status field was
        `noSuchName'."
  ::= { snmp 9 }

snmpInBadValues OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number of SNMP PDUs which were
        delivered to the SNMP protocol entity and for
        which the value of the error-status field was
        `badValue'."
  ::= { snmp 10 }

snmpInReadOnlys OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number valid SNMP PDUs which were delivered
        to the SNMP protocol entity and for which the value
        of the error-status field was `readOnly'.  It should
        be noted that it is a protocol error to generate an
        SNMP PDU which contains the value `readOnly' in the
        error-status field, as such this object is provided
        as a means of detecting incorrect implementations of
        the SNMP."
  ::= { snmp 11 }

snmpInGenErrs OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number of SNMP PDUs which were delivered
        to the SNMP protocol entity and for which the value
        of the error-status field was `genErr'."
  ::= { snmp 12 }

snmpInTotalReqVars OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number of MIB objects which have been
```

```
            retrieved successfully by the SNMP protocol entity
            as the result of receiving valid SNMP Get-Request
            and Get-Next PDUs."
    ::= { snmp 13 }

snmpInTotalSetVars OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      obsolete
    DESCRIPTION
            "The total number of MIB objects which have been
            altered successfully by the SNMP protocol entity as
            the result of receiving valid SNMP Set-Request PDUs."
    ::= { snmp 14 }

snmpInGetRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      obsolete
    DESCRIPTION
            "The total number of SNMP Get-Request PDUs which
            have been accepted and processed by the SNMP
            protocol entity."
    ::= { snmp 15 }

snmpInGetNexts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      obsolete
    DESCRIPTION
            "The total number of SNMP Get-Next PDUs which have been
            accepted and processed by the SNMP protocol entity."
    ::= { snmp 16 }

snmpInSetRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      obsolete
    DESCRIPTION
            "The total number of SNMP Set-Request PDUs which
            have been accepted and processed by the SNMP protocol
            entity."
    ::= { snmp 17 }

snmpInGetResponses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      obsolete
```

```
  DESCRIPTION
          "The total number of SNMP Get-Response PDUs which
          have been accepted and processed by the SNMP protocol
          entity."
  ::= { snmp 18 }

snmpInTraps OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
          "The total number of SNMP Trap PDUs which have been
          accepted and processed by the SNMP protocol entity."
  ::= { snmp 19 }

snmpOutTooBigs OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
          "The total number of SNMP PDUs which were generated
          by the SNMP protocol entity and for which the value
          of the error-status field was `tooBig.'"
  ::= { snmp 20 }

snmpOutNoSuchNames OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
          "The total number of SNMP PDUs which were generated
          by the SNMP protocol entity and for which the value
          of the error-status was `noSuchName'."
  ::= { snmp 21 }

snmpOutBadValues OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
          "The total number of SNMP PDUs which were generated
          by the SNMP protocol entity and for which the value
          of the error-status field was `badValue'."
  ::= { snmp 22 }

-- { snmp 23 } is not used

snmpOutGenErrs OBJECT-TYPE
```

```
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number of SNMP PDUs which were generated
        by the SNMP protocol entity and for which the value
        of the error-status field was `genErr'."
  ::= { snmp 24 }

snmpOutGetRequests OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number of SNMP Get-Request PDUs which
        have been generated by the SNMP protocol entity."
  ::= { snmp 25 }

snmpOutGetNexts OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number of SNMP Get-Next PDUs which have
        been generated by the SNMP protocol entity."
  ::= { snmp 26 }

snmpOutSetRequests OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number of SNMP Set-Request PDUs which
        have been generated by the SNMP protocol entity."
  ::= { snmp 27 }

snmpOutGetResponses OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
  STATUS      obsolete
  DESCRIPTION
        "The total number of SNMP Get-Response PDUs which
        have been generated by the SNMP protocol entity."
  ::= { snmp 28 }

snmpOutTraps OBJECT-TYPE
  SYNTAX      Counter32
  MAX-ACCESS  read-only
```

```
  STATUS      obsolete
  DESCRIPTION
          "The total number of SNMP Trap PDUs which have
          been generated by the SNMP protocol entity."
  ::= { snmp 29 }

snmpObsoleteGroup OBJECT-GROUP
  OBJECTS { snmpOutPkts, snmpInTooBigs, snmpInNoSuchNames,
            snmpInBadValues, snmpInReadOnlys, snmpInGenErrs,
            snmpInTotalReqVars, snmpInTotalSetVars,
            snmpInGetRequests, snmpInGetNexts, snmpInSetRequests,
            snmpInGetResponses, snmpInTraps, snmpOutTooBigs,
            snmpOutNoSuchNames, snmpOutBadValues,
            snmpOutGenErrs, snmpOutGetRequests, snmpOutGetNexts,
            snmpOutSetRequests, snmpOutGetResponses, snmpOutTraps
            }
  STATUS  obsolete
  DESCRIPTION
          "A collection of objects from RFC 1213 made obsolete
          by this MIB module."
  ::= { snmpMIBGroups 10 }

END
```

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/lib/mibs/SNMPv2-MIB.mib
No license file was found, but licenses were detected in source scan.

```
Network Working Group                    Editor of this version:
Request for Comments: 3416                            R. Presuhn
STD: 62                                       BMC Software, Inc.
Obsoletes: 1905                      Authors of previous version:
Category: Standards Track                                J. Case
                                             SNMP Research, Inc.
                                                   K. McCloghrie
                                             Cisco Systems, Inc.
                                                         M. Rose
                                    Dover Beach Consulting, Inc.
                                                   S. Waldbusser
                                  International Network Services
                                                   December 2002


             Version 2 of the Protocol Operations for
           the Simple Network Management Protocol (SNMP)

Status of this Memo
```

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document defines version 2 of the protocol operations for the Simple Network Management Protocol (SNMP). It defines the syntax and elements of procedure for sending, receiving, and processing SNMP PDUs. This document obsoletes RFC 1905.

Table of Contents

1. Introduction

The SNMP Management Framework at the time of this writing consists of five major components:

- An overall architecture, described in STD 62, RFC 3411 [RFC3411].

- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [RFC1155], STD 16, RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [RFC1901] and STD 62, RFC 3417 [RFC3417]. The third version of the message protocol is called SNMPv3 and described in STD 62, RFC 3417 [RFC3417], RFC 3412 [RFC3412] and RFC 3414 [RFC3414].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in this document.

- A set of fundamental applications described in STD 62, RFC 3413 [RFC3413] and the view-based access control mechanism described in STD 62, RFC 3415 [RFC3415].

A more detailed introduction to the SNMP Management Framework at the time of this writing can be found in RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This document, Version 2 of the Protocol Operations for the Simple Network Management Protocol, defines the operations of the protocol with respect to the sending and receiving of PDUs to be carried by the message protocol.

2. Overview

SNMP entities supporting command generator or notification receiver applications (traditionally called "managers") communicate with SNMP entities supporting command responder or notification originator applications (traditionally called "agents"). The purpose of this protocol is the transport of management information and operations.

2.1. Management Information

The term "variable" refers to an instance of a non-aggregate object type defined according to the conventions set forth in the SMI [RFC2578] or the textual conventions based on the SMI [RFC2579]. The term "variable binding" normally refers to the pairing of the name of a variable and its associated value. However, if certain kinds of exceptional conditions occur during processing of a retrieval request, a variable binding will pair a name and an indication of that exception.

A variable-binding list is a simple list of variable bindings.

The name of a variable is an OBJECT IDENTIFIER which is the concatenation of the OBJECT IDENTIFIER of the corresponding object-type together with an OBJECT IDENTIFIER fragment identifying the instance. The OBJECT IDENTIFIER of the corresponding object-type is called the OBJECT IDENTIFIER prefix of the variable.

2.2. Retransmission of Requests

For all types of request in this protocol, the receiver is required under normal circumstances, to generate and transmit a response to the originator of the request. Whether or not a request should be retransmitted if no corresponding response is received in an appropriate time interval, is at the discretion of the application originating the request. This will normally depend on the urgency of the request. However, such an application needs to act responsibly in respect to the frequency and duration of re-transmissions. See BCP 41 [RFC2914] for discussion of relevant congestion control principles.

2.3. Message Sizes

The maximum size of an SNMP message is limited to the minimum of:

(1) the maximum message size which the destination SNMP entity can accept; and,

(2) the maximum message size which the source SNMP entity can generate.

The former may be known on a per-recipient basis; and in the absence of such knowledge, is indicated by transport domain used when sending the message. The latter is imposed by implementation-specific local constraints.

Each transport mapping for the SNMP indicates the minimum message size which a SNMP implementation must be able to produce or consume. Although implementations are encouraged to support larger values whenever possible, a conformant implementation must never generate messages larger than allowed by the receiving SNMP entity.

One of the aims of the GetBulkRequest-PDU, specified in this protocol, is to minimize the number of protocol exchanges required to retrieve a large amount of management information. As such, this PDU type allows an SNMP entity supporting command generator applications to request that the response be as large as possible given the constraints on message sizes. These constraints include the limits on the size of messages which the SNMP entity supporting command responder applications can generate, and the SNMP entity supporting command generator applications can receive.

However, it is possible that such maximum sized messages may be larger than the Path MTU of the path across the network traversed by the messages. In this situation, such messages are subject to fragmentation. Fragmentation is generally considered to be harmful [FRAG], since among other problems, it leads to a decrease in the reliability of the transfer of the messages. Thus, an SNMP entity which sends a GetBulkRequest-PDU must take care to set its parameters accordingly, so as to reduce the risk of fragmentation. In particular, under conditions of network stress, only small values should be used for max-repetitions.

2.4. Transport Mappings

It is important to note that the exchange of SNMP messages requires only an unreliable datagram service, with every message being entirely and independently contained in a single transport datagram. Specific transport mappings and encoding rules are specified elsewhere [RFC3417]. However, the preferred mapping is the use of the User Datagram Protocol [RFC768].

2.5. SMIv2 Data Type Mappings

The SMIv2 [RFC2578] defines 11 base types (INTEGER, OCTET STRING, OBJECT IDENTIFIER, Integer32, IpAddress, Counter32, Gauge32, Unsigned32, TimeTicks, Opaque, Counter64) and the BITS construct. The SMIv2 base types are mapped to the corresponding selection type in the SimpleSyntax and ApplicationSyntax choices of the ASN.1 SNMP protocol definition. Note that the INTEGER and Integer32 SMIv2 base types are mapped to the integer-value selection type of the SimpleSyntax choice. Similarly, the Gauge32 and Unsigned32 SMIv2 base types are mapped to the unsigned-integer-value selection type of the ApplicationSyntax choice.

The SMIv2 BITS construct is mapped to the string-value selection type of the SimpleSyntax choice. A BITS value is encoded as an OCTET STRING, in which all the named bits in (the definition of) the bitstring, commencing with the first bit and proceeding to the last bit, are placed in bits 8 (high order bit) to 1 (low order bit) of the first octet, followed by bits 8 to 1 of each subsequent octet in turn, followed by as many bits as are needed of the final subsequent octet, commencing with bit 8. Remaining bits, if any, of the final octet are set to zero on generation and ignored on receipt.

3. Definitions

The PDU syntax is defined using ASN.1 notation [ASN1].

SNMPv2-PDU DEFINITIONS ::= BEGIN

```
ObjectName ::= OBJECT IDENTIFIER

ObjectSyntax ::= CHOICE {
      simple           SimpleSyntax,
      application-wide ApplicationSyntax }

SimpleSyntax ::= CHOICE {
      integer-value   INTEGER (-2147483648..2147483647),
      string-value    OCTET STRING (SIZE (0..65535)),
      objectID-value  OBJECT IDENTIFIER }

ApplicationSyntax ::= CHOICE {
      ipAddress-value        IpAddress,
      counter-value          Counter32,
      timeticks-value        TimeTicks,
      arbitrary-value        Opaque,
      big-counter-value      Counter64,
      unsigned-integer-value Unsigned32 }
```

Presuhn, et al.   Standards Track   [Page 6]

RFC 3416   Protocol Operations for SNMP   December 2002

```
IpAddress ::= [APPLICATION 0] IMPLICIT OCTET STRING (SIZE (4))

Counter32 ::= [APPLICATION 1] IMPLICIT INTEGER (0..4294967295)

Unsigned32 ::= [APPLICATION 2] IMPLICIT INTEGER (0..4294967295)

Gauge32 ::= Unsigned32

TimeTicks ::= [APPLICATION 3] IMPLICIT INTEGER (0..4294967295)

Opaque ::= [APPLICATION 4] IMPLICIT OCTET STRING

Counter64 ::= [APPLICATION 6]
              IMPLICIT INTEGER (0..18446744073709551615)

-- protocol data units

PDUs ::= CHOICE {
     get-request      GetRequest-PDU,
     get-next-request GetNextRequest-PDU,
```

```
      get-bulk-request GetBulkRequest-PDU,
      response         Response-PDU,
      set-request      SetRequest-PDU,
      inform-request   InformRequest-PDU,
      snmpV2-trap      SNMPv2-Trap-PDU,
      report           Report-PDU }

  -- PDUs

  GetRequest-PDU ::= [0] IMPLICIT PDU

  GetNextRequest-PDU ::= [1] IMPLICIT PDU

  Response-PDU ::= [2] IMPLICIT PDU

  SetRequest-PDU ::= [3] IMPLICIT PDU

  -- [4] is obsolete

  GetBulkRequest-PDU ::= [5] IMPLICIT BulkPDU

  InformRequest-PDU ::= [6] IMPLICIT PDU

  SNMPv2-Trap-PDU ::= [7] IMPLICIT PDU

  --   Usage and precise semantics of Report-PDU are not defined
  --   in this document.  Any SNMP administrative framework making
  --   use of this PDU must define its usage and semantics.
```



```
  Report-PDU ::= [8] IMPLICIT PDU

  max-bindings INTEGER ::= 2147483647

  PDU ::= SEQUENCE {
          request-id INTEGER (-214783648..214783647),

          error-status                -- sometimes ignored
              INTEGER {
                  noError(0),
                  tooBig(1),
                  noSuchName(2),      -- for proxy compatibility
```

```
            badValue(3),        -- for proxy compatibility
            readOnly(4),        -- for proxy compatibility
            genErr(5),
            noAccess(6),
            wrongType(7),
            wrongLength(8),
            wrongEncoding(9),
            wrongValue(10),
            noCreation(11),
            inconsistentValue(12),
            resourceUnavailable(13),
            commitFailed(14),
            undoFailed(15),
            authorizationError(16),
            notWritable(17),
            inconsistentName(18)
        },

      error-index               -- sometimes ignored
        INTEGER (0..max-bindings),

      variable-bindings         -- values are sometimes ignored
        VarBindList
    }

  BulkPDU ::=                   -- must be identical in
    SEQUENCE {                  -- structure to PDU
      request-id      INTEGER (-214783648..214783647),
      non-repeaters   INTEGER (0..max-bindings),
      max-repetitions INTEGER (0..max-bindings),

      variable-bindings         -- values are ignored
        VarBindList
    }

  -- variable binding
```

Presuhn, et al.   Standards Track   [Page 8]

RFC 3416   Protocol Operations for SNMP   December 2002

```
  VarBind ::= SEQUENCE {
      name ObjectName,

      CHOICE {
```

```
            value         ObjectSyntax,
            unSpecified    NULL,    -- in retrieval requests

                            -- exceptions in responses
            noSuchObject   [0] IMPLICIT NULL,
            noSuchInstance [1] IMPLICIT NULL,
            endOfMibView   [2] IMPLICIT NULL
        }
    }

-- variable-binding list

VarBindList ::= SEQUENCE (SIZE (0..max-bindings)) OF VarBind

END
```

4. Protocol Specification

4.1. Common Constructs

The value of the request-id field in a Response-PDU takes the value of the request-id field in the request PDU to which it is a response. By use of the request-id value, an application can distinguish the (potentially multiple) outstanding requests, and thereby correlate incoming responses with outstanding requests. In cases where an unreliable datagram service is used, the request-id also provides a simple means of identifying messages duplicated by the network. Use of the same request-id on a retransmission of a request allows the response to either the original transmission or the retransmission to satisfy the request. However, in order to calculate the round trip time for transmission and processing of a request-response transaction, the application needs to use a different request-id value on a retransmitted request. The latter strategy is recommended for use in the majority of situations.

A non-zero value of the error-status field in a Response-PDU is used to indicate that an error occurred to prevent the processing of the request. In these cases, a non-zero value of the Response-PDU's error-index field provides additional information by identifying which variable binding in the list caused the error. A variable binding is identified by its index value. The first variable binding in a variable-binding list is index one, the second is index two, etc.

SNMP limits OBJECT IDENTIFIER values to a maximum of 128 sub-identifiers, where each sub-identifier has a maximum value of 2**32-1.

4.2. PDU Processing

In the elements of procedure below, any field of a PDU which is not referenced by the relevant procedure is ignored by the receiving SNMP entity. However, all components of a PDU, including those whose values are ignored by the receiving SNMP entity, must have valid ASN.1 syntax and encoding. For example, some PDUs (e.g., the GetRequest-PDU) are concerned only with the name of a variable and not its value. In this case, the value portion of the variable binding is ignored by the receiving SNMP entity. The unSpecified value is defined for use as the value portion of such bindings.

On generating a management communication, the message "wrapper" to encapsulate the PDU is generated according to the "Elements of Procedure" of the administrative framework in use. The definition of "max-bindings" imposes an upper bound on the number of variable bindings. In practice, the size of a message is also limited by constraints on the maximum message size. A compliant implementation must support as many variable bindings in a PDU or BulkPDU as fit into the overall maximum message size limit of the SNMP engine, but no more than 2147483647 variable bindings.

On receiving a management communication, the "Elements of Procedure" of the administrative framework in use is followed, and if those procedures indicate that the operation contained within the message is to be performed locally, then those procedures also indicate the MIB view which is visible to the operation.

4.2.1. The GetRequest-PDU

A GetRequest-PDU is generated and transmitted at the request of an application.

Upon receipt of a GetRequest-PDU, the receiving SNMP entity processes each variable binding in the variable-binding list to produce a Response-PDU. All fields of the Response-PDU have the same values as the corresponding fields of the received request except as indicated below. Each variable binding is processed as follows:

(1) If the variable binding's name exactly matches the name of a

variable accessible by this request, then the variable
binding's value field is set to the value of the named
variable.

(2) Otherwise, if the variable binding's name does not have an OBJECT IDENTIFIER prefix which exactly matches the OBJECT IDENTIFIER prefix of any (potential) variable accessible by this request, then its value field is set to "noSuchObject".

(3) Otherwise, the variable binding's value field is set to "noSuchInstance".

If the processing of any variable binding fails for a reason other than listed above, then the Response-PDU is re-formatted with the same values in its request-id and variable-bindings fields as the received GetRequest-PDU, with the value of its error-status field set to "genErr", and the value of its error-index field is set to the index of the failed variable binding.

Otherwise, the value of the Response-PDU's error-status field is set to "noError", and the value of its error-index field is zero.

The generated Response-PDU is then encapsulated into a message. If the size of the resultant message is less than or equal to both a local constraint and the maximum message size of the originator, it is transmitted to the originator of the GetRequest-PDU.

Otherwise, an alternate Response-PDU is generated. This alternate Response-PDU is formatted with the same value in its request-id field as the received GetRequest-PDU, with the value of its error-status field set to "tooBig", the value of its error-index field set to zero, and an empty variable-bindings field. This alternate Response-PDU is then encapsulated into a message. If the size of the resultant message is less than or equal to both a local constraint and the maximum message size of the originator, it is transmitted to the originator of the GetRequest-PDU. Otherwise, the snmpSilentDrops [RFC3418] counter is incremented and the resultant message is discarded.

4.2.2. The GetNextRequest-PDU

A GetNextRequest-PDU is generated and transmitted at the request of an application.

Upon receipt of a GetNextRequest-PDU, the receiving SNMP entity processes each variable binding in the variable-binding list to produce a Response-PDU. All fields of the Response-PDU have the same values as the corresponding fields of the received request except as indicated below. Each variable binding is processed as follows:

(1) The variable is located which is in the lexicographically ordered list of the names of all variables which are

accessible by this request and whose name is the first lexicographic successor of the variable binding's name in the incoming GetNextRequest-PDU. The corresponding variable binding's name and value fields in the Response-PDU are set to the name and value of the located variable.

(2) If the requested variable binding's name does not lexicographically precede the name of any variable accessible by this request, i.e., there is no lexicographic successor, then the corresponding variable binding produced in the Response-PDU has its value field set to "endOfMibView", and its name field set to the variable binding's name in the request.

If the processing of any variable binding fails for a reason other than listed above, then the Response-PDU is re-formatted with the same values in its request-id and variable-bindings fields as the received GetNextRequest-PDU, with the value of its error-status field set to "genErr", and the value of its error-index field is set to the index of the failed variable binding.

Otherwise, the value of the Response-PDU's error-status field is set to "noError", and the value of its error-index field is zero.

The generated Response-PDU is then encapsulated into a message. If the size of the resultant message is less than or equal to both a local constraint and the maximum message size of the originator, it is transmitted to the originator of the GetNextRequest-PDU.

Otherwise, an alternate Response-PDU is generated. This alternate Response-PDU is formatted with the same values in its request-id field as the received GetNextRequest-PDU, with the value of its error-status field set to "tooBig", the value of its error-index field set to zero, and an empty variable-bindings field. This alternate Response-PDU is then encapsulated into a message. If the size of the resultant message is less than or equal to both a local constraint and the maximum message size of the originator, it is transmitted to the originator of the GetNextRequest-PDU. Otherwise, the snmpSilentDrops [RFC3418] counter is incremented and the resultant message is discarded.

4.2.2.1. Example of Table Traversal

An important use of the GetNextRequest-PDU is the traversal of conceptual tables of information within a MIB. The semantics of this type of request, together with the method of identifying individual instances of objects in the MIB, provides access to related objects in the MIB as if they enjoyed a tabular organization.

In the protocol exchange sketched below, an application retrieves the media-dependent physical address and the address-mapping type for each entry in the IP net-to-media Address Translation Table [RFC1213] of a particular network element. It also retrieves the value of sysUpTime [RFC3418], at which the mappings existed. Suppose that the command responder's IP net-to-media table has three entries:

Interface-Number	Network-Address	Physical-Address	Type
1	10.0.0.51	00:00:10:01:23:45	static
1	9.2.3.4	00:00:10:54:32:10	dynamic
2	10.0.0.15	00:00:10:98:76:54	dynamic

The SNMP entity supporting a command generator application begins by sending a GetNextRequest-PDU containing the indicated OBJECT IDENTIFIER values as the requested variable names:

```
GetNextRequest ( sysUpTime,
                 ipNetToMediaPhysAddress,
                 ipNetToMediaType )
```

The SNMP entity supporting a command responder application responds with a Response-PDU:

```
Response (( sysUpTime.0 =  "123456" ),
          ( ipNetToMediaPhysAddress.1.9.2.3.4 = "000010543210" ),
        ( ipNetToMediaType.1.9.2.3.4 =  "dynamic" ))
```

The SNMP entity supporting the command generator application continues with:

```
GetNextRequest ( sysUpTime,
              ipNetToMediaPhysAddress.1.9.2.3.4,
              ipNetToMediaType.1.9.2.3.4 )
```

The SNMP entity supporting the command responder application responds with:

```
Response (( sysUpTime.0 =  "123461" ),
          ( ipNetToMediaPhysAddress.1.10.0.0.51 = "000010012345" ),
        ( ipNetToMediaType.1.10.0.0.51 =  "static" ))
```

The SNMP entity supporting the command generator application continues with:

```
GetNextRequest ( sysUpTime,
              ipNetToMediaPhysAddress.1.10.0.0.51,
              ipNetToMediaType.1.10.0.0.51 )
```


The SNMP entity supporting the command responder application responds with:

```
Response (( sysUpTime.0 =  "123466" ),
          ( ipNetToMediaPhysAddress.2.10.0.0.15 = "000010987654" ),
        ( ipNetToMediaType.2.10.0.0.15 =  "dynamic" ))
```

The SNMP entity supporting the command generator application continues with:

```
GetNextRequest ( sysUpTime,
              ipNetToMediaPhysAddress.2.10.0.0.15,
```

ipNetToMediaType.2.10.0.0.15)

As there are no further entries in the table, the SNMP entity supporting the command responder application responds with the variables that are next in the lexicographical ordering of the accessible object names, for example:

```
Response (( sysUpTime.0 =  "123471" ),
          ( ipNetToMediaNetAddress.1.9.2.3.4 = "9.2.3.4" ),
        ( ipRoutingDiscards.0 =  "2" ))
```

Note how, having reached the end of the column for ipNetToMediaPhysAddress, the second variable binding from the command responder application has now "wrapped" to the first row in the next column. Furthermore, note how, having reached the end of the ipNetToMediaTable for the third variable binding, the command responder application has responded with the next available object, which is outside that table. This response signals the end of the table to the command generator application.

4.2.3. The GetBulkRequest-PDU

A GetBulkRequest-PDU is generated and transmitted at the request of an application. The purpose of the GetBulkRequest-PDU is to request the transfer of a potentially large amount of data, including, but not limited to, the efficient and rapid retrieval of large tables.

Upon receipt of a GetBulkRequest-PDU, the receiving SNMP entity processes each variable binding in the variable-binding list to produce a Response-PDU with its request-id field having the same value as in the request.

For the GetBulkRequest-PDU type, the successful processing of each variable binding in the request generates zero or more variable bindings in the Response-PDU. That is, the one-to-one mapping between the variable bindings of the GetRequest-PDU, GetNextRequest-

PDU, and SetRequest-PDU types and the resultant Response-PDUs does not apply for the mapping between the variable bindings of a GetBulkRequest-PDU and the resultant Response-PDU.

The values of the non-repeaters and max-repetitions fields in the request specify the processing requested. One variable binding in the Response-PDU is requested for the first N variable bindings in the request and M variable bindings are requested for each of the R remaining variable bindings in the request. Consequently, the total number of requested variable bindings communicated by the request is given by N + (M * R), where N is the minimum of: a) the value of the non-repeaters field in the request, and b) the number of variable bindings in the request; M is the value of the max-repetitions field in the request; and R is the maximum of: a) number of variable bindings in the request - N, and b) zero.

The receiving SNMP entity produces a Response-PDU with up to the total number of requested variable bindings communicated by the request. The request-id shall have the same value as the received GetBulkRequest-PDU.

If N is greater than zero, the first through the (N)-th variable bindings of the Response-PDU are each produced as follows:

(1) The variable is located which is in the lexicographically ordered list of the names of all variables which are accessible by this request and whose name is the first lexicographic successor of the variable binding's name in the incoming GetBulkRequest-PDU. The corresponding variable binding's name and value fields in the Response-PDU are set to the name and value of the located variable.

(2) If the requested variable binding's name does not lexicographically precede the name of any variable accessible by this request, i.e., there is no lexicographic successor, then the corresponding variable binding produced in the Response-PDU has its value field set to "endOfMibView", and its name field set to the variable binding's name in the request.

If M and R are non-zero, the (N + 1)-th and subsequent variable bindings of the Response-PDU are each produced in a similar manner. For each iteration i, such that i is greater than zero and less than or equal to M, and for each repeated variable, r, such that r is greater than zero and less than or equal to R, the (N + ((i-1) * R) + r)-th variable binding of the Response-PDU is produced as follows:

(1) The variable which is in the lexicographically ordered list of the names of all variables which are accessible by this request and whose name is the (i)-th lexicographic successor of the (N + r)-th variable binding's name in the incoming GetBulkRequest-PDU is located and the variable binding's name and value fields are set to the name and value of the located variable.

(2) If there is no (i)-th lexicographic successor, then the corresponding variable binding produced in the Response-PDU has its value field set to "endOfMibView", and its name field set to either the last lexicographic successor, or if there are no lexicographic successors, to the (N + r)-th variable binding's name in the request.

While the maximum number of variable bindings in the Response-PDU is bounded by N + (M * R), the response may be generated with a lesser number of variable bindings (possibly zero) for either of three reasons.

(1) If the size of the message encapsulating the Response-PDU containing the requested number of variable bindings would be greater than either a local constraint or the maximum message size of the originator, then the response is generated with a lesser number of variable bindings. This lesser number is the ordered set of variable bindings with some of the variable bindings at the end of the set removed, such that the size of the message encapsulating the Response-PDU is approximately equal to but no greater than either a local constraint or the maximum message size of the originator. Note that the number of variable bindings removed has no relationship to the values of N, M, or R.

(2) The response may also be generated with a lesser number of variable bindings if for some value of iteration i, such that i is greater than zero and less than or equal to M, that all of the generated variable bindings have the value field set to "endOfMibView". In this case, the variable bindings may be truncated after the (N + (i * R))-th variable binding.

(3) In the event that the processing of a request with many repetitions requires a significantly greater amount of processing time than a normal request, then a command responder application may terminate the request with less than the full

number of repetitions, providing at least one repetition is
completed.

If the processing of any variable binding fails for a reason other than listed above, then the Response-PDU is re-formatted with the same values in its request-id and variable-bindings fields as the received GetBulkRequest-PDU, with the value of its error-status field set to "genErr", and the value of its error-index field is set to the index of the variable binding in the original request which corresponds to the failed variable binding.

Otherwise, the value of the Response-PDU's error-status field is set to "noError", and the value of its error-index field to zero.

The generated Response-PDU (possibly with an empty variable-bindings field) is then encapsulated into a message. If the size of the resultant message is less than or equal to both a local constraint and the maximum message size of the originator, it is transmitted to the originator of the GetBulkRequest-PDU. Otherwise, the snmpSilentDrops [RFC3418] counter is incremented and the resultant message is discarded.

4.2.3.1. Another Example of Table Traversal

This example demonstrates how the GetBulkRequest-PDU can be used as an alternative to the GetNextRequest-PDU. The same traversal of the IP net-to-media table as shown in Section 4.2.2.1 is achieved with fewer exchanges.

The SNMP entity supporting the command generator application begins by sending a GetBulkRequest-PDU with the modest max-repetitions value of 2, and containing the indicated OBJECT IDENTIFIER values as the requested variable names:

```
GetBulkRequest [ non-repeaters = 1, max-repetitions = 2 ]
              ( sysUpTime,
                ipNetToMediaPhysAddress,
                ipNetToMediaType )
```

The SNMP entity supporting the command responder application responds with a Response-PDU:

```
 Response (( sysUpTime.0 =  "123456" ),
           ( ipNetToMediaPhysAddress.1.9.2.3.4 = "000010543210" ),
         ( ipNetToMediaType.1.9.2.3.4 =  "dynamic" ),
           ( ipNetToMediaPhysAddress.1.10.0.0.51 = "000010012345" ),
         ( ipNetToMediaType.1.10.0.0.51 =  "static" ))
```


The SNMP entity supporting the command generator application continues with:

```
  GetBulkRequest [ non-repeaters = 1, max-repetitions = 2 ]
                ( sysUpTime,
                  ipNetToMediaPhysAddress.1.10.0.0.51,
                  ipNetToMediaType.1.10.0.0.51 )
```

The SNMP entity supporting the command responder application responds with:

```
 Response (( sysUpTime.0 =  "123466" ),
           ( ipNetToMediaPhysAddress.2.10.0.0.15 = "000010987654" ),
           ( ipNetToMediaType.2.10.0.0.15 = "dynamic" ),
           ( ipNetToMediaNetAddress.1.9.2.3.4 = "9.2.3.4" ),
         ( ipRoutingDiscards.0 =  "2" ))
```

Note how, as in the first example, the variable bindings in the response indicate that the end of the table has been reached. The fourth variable binding does so by returning information from the next available column; the fifth variable binding does so by returning information from the first available object lexicographically following the table. This response signals the end of the table to the command generator application.

4.2.4. The Response-PDU

The Response-PDU is generated by an SNMP entity only upon receipt of a GetRequest-PDU, GetNextRequest-PDU, GetBulkRequest-PDU, SetRequest-PDU, or InformRequest-PDU, as described elsewhere in this document.

If the error-status field of the Response-PDU is non-zero, the value fields of the variable bindings in the variable binding list are ignored.

If both the error-status field and the error-index field of the Response-PDU are non-zero, then the value of the error-index field is the index of the variable binding (in the variable-binding list of the corresponding request) for which the request failed. The first variable binding in a request's variable-binding list is index one, the second is index two, etc.

A compliant SNMP entity supporting a command generator application must be able to properly receive and handle a Response-PDU with an error-status field equal to "noSuchName", "badValue", or "readOnly". (See sections 1.3 and 4.3 of [RFC2576].)

Upon receipt of a Response-PDU, the receiving SNMP entity presents its contents to the application which generated the request with the same request-id value. For more details, see [RFC3412].

4.2.5. The SetRequest-PDU

A SetRequest-PDU is generated and transmitted at the request of an application.

Upon receipt of a SetRequest-PDU, the receiving SNMP entity determines the size of a message encapsulating a Response-PDU having the same values in its request-id and variable-bindings fields as the received SetRequest-PDU, and the largest possible sizes of the error-status and error-index fields. If the determined message size is greater than either a local constraint or the maximum message size of the originator, then an alternate Response-PDU is generated, transmitted to the originator of the SetRequest-PDU, and processing of the SetRequest-PDU terminates immediately thereafter. This alternate Response-PDU is formatted with the same values in its request-id field as the received SetRequest-PDU, with the value of its error-status field set to "tooBig", the value of its error-index field set to zero, and an empty variable-bindings field. This alternate Response-PDU is then encapsulated into a message. If the size of the resultant message is less than or equal to both a local constraint and the maximum message size of the originator, it is transmitted to the originator of the SetRequest-PDU. Otherwise, the snmpSilentDrops [RFC3418] counter is incremented and the resultant message is discarded. Regardless, processing of the SetRequest-PDU terminates.

Otherwise, the receiving SNMP entity processes each variable binding in the variable-binding list to produce a Response-PDU. All fields of the Response-PDU have the same values as the corresponding fields of the received request except as indicated below.

The variable bindings are conceptually processed as a two phase operation. In the first phase, each variable binding is validated; if all validations are successful, then each variable is altered in the second phase. Of course, implementors are at liberty to implement either the first, or second, or both, of these conceptual phases as multiple implementation phases. Indeed, such multiple implementation phases may be necessary in some cases to ensure consistency.

The following validations are performed in the first phase on each variable binding until they are all successful, or until one fails:

(1) If the variable binding's name specifies an existing or non-existent variable to which this request is/would be denied access because it is/would not be in the appropriate MIB view, then the value of the Response-PDU's error-status field is set to "noAccess", and the value of its error-index field is set to the index of the failed variable binding.

(2) Otherwise, if there are no variables which share the same OBJECT IDENTIFIER prefix as the variable binding's name, and

which are able to be created or modified no matter what new value is specified, then the value of the Response-PDU's error-status field is set to "notWritable", and the value of its error-index field is set to the index of the failed variable binding.

(3) Otherwise, if the variable binding's value field specifies, according to the ASN.1 language, a type which is inconsistent with that required for all variables which share the same OBJECT IDENTIFIER prefix as the variable binding's name, then the value of the Response-PDU's error-status field is set to "wrongType", and the value of its error-index field is set to the index of the failed variable binding.

(4) Otherwise, if the variable binding's value field specifies, according to the ASN.1 language, a length which is inconsistent with that required for all variables which share the same OBJECT IDENTIFIER prefix as the variable binding's name, then the value of the Response-PDU's error-status field is set to "wrongLength", and the value of its error-index field is set to the index of the failed variable binding.

(5) Otherwise, if the variable binding's value field contains an ASN.1 encoding which is inconsistent with that field's ASN.1 tag, then the value of the Response-PDU's error-status field is set to "wrongEncoding", and the value of its error-index field is set to the index of the failed variable binding. (Note that not all implementation strategies will generate this error.)

(6) Otherwise, if the variable binding's value field specifies a value which could under no circumstances be assigned to the variable, then the value of the Response-PDU's error-status field is set to "wrongValue", and the value of its error-index field is set to the index of the failed variable binding.

(7) Otherwise, if the variable binding's name specifies a variable which does not exist and could not ever be created (even though some variables sharing the same OBJECT IDENTIFIER prefix might under some circumstances be able to be created), then the value

of the Response-PDU's error-status field is set to "noCreation", and the value of its error-index field is set to the index of the failed variable binding.

(8) Otherwise, if the variable binding's name specifies a variable which does not exist but can not be created under the present circumstances (even though it could be created under other circumstances), then the value of the Response-PDU's error-status field is set to "inconsistentName", and the value of its error-index field is set to the index of the failed variable binding.

(9) Otherwise, if the variable binding's name specifies a variable which exists but can not be modified no matter what new value is specified, then the value of the Response-PDU's error-status field is set to "notWritable", and the value of its error-index field is set to the index of the failed variable binding.

(10) Otherwise, if the variable binding's value field specifies a value that could under other circumstances be held by the variable, but is presently inconsistent or otherwise unable to be assigned to the variable, then the value of the Response-PDU's error-status field is set to "inconsistentValue", and the value of its error-index field is set to the index of the failed variable binding.

(11) When, during the above steps, the assignment of the value specified by the variable binding's value field to the specified variable requires the allocation of a resource which is presently unavailable, then the value of the Response-PDU's error-status field is set to "resourceUnavailable", and the value of its error-index field is set to the index of the failed variable binding.

(12) If the processing of the variable binding fails for a reason other than listed above, then the value of the Response-PDU's error-status field is set to "genErr", and the value of its error-index field is set to the index of the failed variable binding.

(13) Otherwise, the validation of the variable binding succeeds.

At the end of the first phase, if the validation of all variable bindings succeeded, then the value of the Response-PDU's error-status field is set to "noError" and the value of its error-index field is zero, and processing continues as follows.

For each variable binding in the request, the named variable is created if necessary, and the specified value is assigned to it. Each of these variable assignments occurs as if simultaneously with respect to all other assignments specified in the same request. However, if the same variable is named more than once in a single request, with different associated values, then the actual assignment made to that variable is implementation-specific.

If any of these assignments fail (even after all the previous validations), then all other assignments are undone, and the Response-PDU is modified to have the value of its error-status field set to "commitFailed", and the value of its error-index field set to the index of the failed variable binding.

If and only if it is not possible to undo all the assignments, then the Response-PDU is modified to have the value of its error-status field set to "undoFailed", and the value of its error-index field is set to zero. Note that implementations are strongly encouraged to take all possible measures to avoid use of either "commitFailed" or "undoFailed" - these two error-status codes are not to be taken as license to take the easy way out in an implementation.

Finally, the generated Response-PDU is encapsulated into a message, and transmitted to the originator of the SetRequest-PDU.

4.2.6. The SNMPv2-Trap-PDU

An SNMPv2-Trap-PDU is generated and transmitted by an SNMP entity on behalf of a notification originator application. The SNMPv2-Trap-PDU is often used to notify a notification receiver application at a logically remote SNMP entity that an event has occurred or that a condition is present. There is no confirmation associated with this notification delivery mechanism.

The destination(s) to which an SNMPv2-Trap-PDU is sent is determined in an implementation-dependent fashion by the SNMP entity. The first two variable bindings in the variable binding list of an SNMPv2-Trap-PDU are sysUpTime.0 [RFC3418] and snmpTrapOID.0 [RFC3418] respectively. If the OBJECTS clause is present in the invocation of the corresponding NOTIFICATION-TYPE macro, then each corresponding variable, as instantiated by this notification, is copied, in order,

to the variable-bindings field. If any additional variables are being included (at the option of the generating SNMP entity), then each is copied to the variable-bindings field.

4.2.7. The InformRequest-PDU

An InformRequest-PDU is generated and transmitted by an SNMP entity on behalf of a notification originator application. The InformRequest-PDU is often used to notify a notification receiver application that an event has occurred or that a condition is present. This is a confirmed notification delivery mechanism, although there is, of course, no guarantee of delivery.

The destination(s) to which an InformRequest-PDU is sent is specified by the notification originator application. The first two variable bindings in the variable binding list of an InformRequest-PDU are sysUpTime.0 [RFC3418] and snmpTrapOID.0 [RFC3418] respectively. If the OBJECTS clause is present in the invocation of the corresponding NOTIFICATION-TYPE macro, then each corresponding variable, as instantiated by this notification, is copied, in order, to the variable-bindings field. If any additional variables are being included (at the option of the generating SNMP entity), then each is copied to the variable-bindings field.

Upon receipt of an InformRequest-PDU, the receiving SNMP entity determines the size of a message encapsulating a Response-PDU with the same values in its request-id, error-status, error-index and variable-bindings fields as the received InformRequest-PDU. If the determined message size is greater than either a local constraint or the maximum message size of the originator, then an alternate Response-PDU is generated, transmitted to the originator of the InformRequest-PDU, and processing of the InformRequest-PDU terminates immediately thereafter. This alternate Response-PDU is formatted with the same values in its request-id field as the received InformRequest-PDU, with the value of its error-status field set to "tooBig", the value of its error-index field set to zero, and an empty variable-bindings field. This alternate Response-PDU is then encapsulated into a message. If the size of the resultant message is less than or equal to both a local constraint and the maximum message size of the originator, it is transmitted to the originator of the InformRequest-PDU. Otherwise, the snmpSilentDrops [RFC3418] counter is incremented and the resultant message is discarded. Regardless, processing of the InformRequest-PDU terminates.

Otherwise, the receiving SNMP entity:

(1) presents its contents to the appropriate application;

(2) generates a Response-PDU with the same values in its request-id and variable-bindings fields as the received InformRequest-PDU, with the value of its error-status field set to "noError" and the value of its error-index field set to zero; and

(3) transmits the generated Response-PDU to the originator of the InformRequest-PDU.

5. Notice on Intellectual Property

The IETF 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 this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive

Director.

6. Acknowledgments

This document is the product of the SNMPv3 Working Group. Some special thanks are in order to the following Working Group members:

Randy Bush
Jeffrey D. Case
Mike Daniele
Rob Frye
Lauren Heintz
Keith McCloghrie
Russ Mundy
David T. Perkins
Randy Presuhn
Aleksey Romanov
Juergen Schoenwaelder
Bert Wijnen

This version of the document, edited by Randy Presuhn, was initially based on the work of a design team whose members were:

Jeffrey D. Case
Keith McCloghrie
David T. Perkins
Randy Presuhn
Juergen Schoenwaelder

The previous versions of this document, edited by Keith McCloghrie, was the result of significant work by four major contributors:

Jeffrey D. Case
Keith McCloghrie
Marshall T. Rose
Steven Waldbusser

Additionally, the contributions of the SNMPv2 Working Group to the previous versions are also acknowledged. In particular, a special thanks is extended for the contributions of:

Alexander I. Alten
Dave Arneson
Uri Blumenthal
Doug Book
Kim Curran
Jim Galvin
Maria Greene
Iain Hanson
Dave Harrington
Nguyen Hien
Jeff Johnson
Michael Kornegay
Deirdre Kostick
David Levi
Daniel Mahoney
Bob Natale
Brian O'Keefe
Andrew Pearson
Dave Perkins
Randy Presuhn
Aleksey Romanov
Shawn Routhier
Jon Saperia
Juergen Schoenwaelder
Bob Stewart

Kaj Tesink
Glenn Waters
Bert Wijnen

7. Security Considerations

The protocol defined in this document by itself does not provide a secure environment. Even if the network itself is secure (for example by using IPSec), there is no control as to who on the secure network is allowed access to management information.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model STD 62, RFC 3414 [RFC3414] and the View-based Access Control Model STD 62, RFC 3415 [RFC3415] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity is properly configured so that:

- only those principals (users) having legitimate rights can access or modify the values of any MIB objects supported by that entity;

- the occurrence of particular events on the entity will be communicated appropriately;

- the entity responds appropriately and with due credence to events and information that have been communicated to it.

8. References

8.1. Normative References

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3411] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3412] Case, J., Harrington, D., Presuhn, R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, December 2002.

[RFC3413] Levi, D., Meyer, P. and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.

[RFC3414] Blumenthal, U. and B. Wijnen, "The User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3415] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

[RFC3417] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for the Simple Network Management Protocol", STD 62, RFC 3417, December 2002.

[RFC3418] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[ASN1] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, December 1987.

8.2. Informative References

[FRAG] Kent, C. and J. Mogul, "Fragmentation Considered Harmful," Proceedings, ACM SIGCOMM '87, Stowe, VT, August 1987.

[RFC1155] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[RFC1157] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[RFC1212] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[RFC1213] McCloghrie, K. and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, March 1991.

[RFC1215] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[RFC1901] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[RFC2576] Frye, R., Levi, D., Routhier, S. and B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework", RFC 2576, March 2000.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

9. Changes from RFC 1905

These are the changes from RFC 1905:

- Corrected spelling error in copyright statement;

- Updated copyright date;

- Updated with new editor's name and contact information;

- Added notice on intellectual property;

- Cosmetic fixes to layout and typography;

- Added table of contents;

- Title changed;

- Updated document headers and footers;

- Deleted the old clause 2.3, entitled "Access to Management Information";

- Changed the way in which request-id was defined, though with the same ultimate syntax and semantics, to avoid coupling with SMI. This does not affect the protocol in any way;

- Replaced the word "exception" with the word "error" in the old clause 4.1. This does not affect the protocol in any way;

- Deleted the first two paragraphs of the old clause 4.2;

- Clarified the maximum number of variable bindings that an implementation must support in a PDU. This does not affect the protocol in any way;

- Replaced occurrences of "SNMPv2 application" with "application";

- Deleted three sentences in old clause 4.2.3 describing the handling of an impossible situation. This does not affect the protocol in any way;

- Clarified the use of the SNMPv2-Trap-Pdu in the old clause 4.2.6. This does not affect the protocol in any way;

- Aligned description of the use of the InformRequest-Pdu in old clause 4.2.7 with the architecture. This does not affect the protocol in any way;

- Updated references;

- Re-wrote introduction clause;

- Replaced manager/agent/SNMPv2 entity terminology with terminology from RFC 2571. This does not affect the protocol in any way;

- Eliminated IMPORTS from the SMI, replaced with equivalent in-line ASN.1. This does not affect the protocol in any way;

- Added notes calling attention to two different manifestations of reaching the end of a table in the table walk examples;

- Added content to security considerations clause;

- Updated ASN.1 comment on use of Report-PDU. This does not affect the protocol in any way;

- Updated acknowledgments section;

- Included information on handling of BITS;

- Deleted spurious comma in ASN.1 definition of PDUs;

- Added abstract;

- Made handling of additional variable bindings in informs consistent with that for traps. This was a correction of an editorial oversight, and reflects implementation practice;

- Added reference to RFC 2914.

10. Editor's Address

Randy Presuhn
BMC Software, Inc.
2141 North First Street
San Jose, CA 95131

USA

Phone: +1 408 546 1006
EMail: randy_presuhn@bmc.com

11. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the
Internet Society.

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/ref/rfc/v3/rfc3416.txt
No license file was found, but licenses were detected in source scan.

net-snmp

This module implements versions 1, 2c and 3 of the [Simple Network Management
Protocol (SNMP)][SNMP].

This module is installed using [node package manager (npm)][npm]:

```
npm install net-snmp
```

It is loaded using the `require()` function:

```
var snmp = require ("net-snmp");
```

Sessions to remote hosts can then be created and used to perform SNMP requests
and send SNMP traps or informs:

```
var session = snmp.createSession ("127.0.0.1", "public");

var oids = ["1.3.6.1.2.1.1.5.0", "1.3.6.1.2.1.1.6.0"];

session.get (oids, function (error, varbinds) {
    if (error) {
        console.error (error);
    } else {
        for (var i = 0; i < varbinds.length; i++)
            if (snmp.isVarbindError (varbinds[i]))
                console.error (snmp.varbindError (varbinds[i]))
            else
                console.log (varbinds[i].oid + " = " + varbinds[i].value);
    }
    session.close ();
});

session.trap (snmp.TrapType.LinkDown, function (error) {
    if (error)
        console.error (error);
});
```

[SNMP]: http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol "SNMP"
[npm]: https://npmjs.org/ "npm"

Applications

RFC 3413 describes five types of SNMP applications:

1. Command Generator Applications — which initiate read or write requests
2. Command Responder Applications — which respond to received read or write requests
3. Notification Originator Applications — which generate notifications (traps or informs)
4. Notification Receiver Applications — which receive notifications (traps or informs)
5. Proxy Forwarder Applications — which forward SNMP messages

This library provides support for all of the above applications, with the documentation
for each shown in this table:

Application	Common Use	Documentation
Command Generator	NMS / SNMP tools	[Using This Module: Command & Notification Generator](#using-this-module-command--notification-generator)
Command Responder	SNMP agents	[Using This Module: SNMP Agent](#using-this-module-snmp-agent)

Notification Originator	SNMP agents / NMS-to-NMS notifications	[Using This Module: Command & Notification Generator](#using-this-module-command--notification-generator)
Notification Receiver	NMS	[Using This Module: Notification Receiver](#using-this-module-notification-receiver)
Proxy Forwarder	SNMP agents	[Forwarder Module](#forwarder-module)

Features

* Support for all SNMP versions: SNMPv1, SNMPv2c and SNMPv3
* SNMPv3 message authentication using MD5 or SHA, and privacy using DES or AES encryption
* Community-based and user-based authorization
* SNMP initiator for all relevant protocol operations: Get, GetNext, GetBulk, Set, Trap, Inform
* Convenience methods for MIB "walking", subtree collection, table and table column collection
* SNMPv3 context support
* Notification receiver for traps and informs
* MIB parsing and MIB module store
* SNMP agent with MIB management for both scalar and tabular data
* Agent table index support for non-integer keys, foreign keys, composite keys and table augmentation
* SNMP proxy forwarder for agent
* AgentX subagent
* IPv4 and IPv6

Standards Compliance

This module aims to be fully compliant with the following RFCs:

* [1098][1098] - A Simple Network Management Protocol (version 1)
* [1155][1155] - Structure and Identification of Management Information
* [2571][2571] - Agent Extensibility (AgentX) Protocol Version 1
* [2578][2578] - Structure of Management Information Version 2 (SMIv2)
* [3413][3413] - Simple Network Management Protocol (SNMP) Applications
* [3414][3414] - User-based Security Model (USM) for version 3 of the
Simple Network Management Protocol (SNMPv3)
* [3416][3416] - Version 2 of the Protocol Operations for the Simple
Network Management Protocol (SNMP)
* [3417][3417] - Transport Mappings for the Simple Network Management
Protocol (SNMP)
* [3826][3826] - The Advanced Encryption Standard (AES) Cipher Algorithm
in the SNMP User-based Security Model

[1155]: https://tools.ietf.org/rfc/rfc1155.txt "RFC 1155"
[1098]: https://tools.ietf.org/rfc/rfc1098.txt "RFC 1098"
[2571]: https://tools.ietf.org/rfc/rfc2578.txt "RFC 2571"
[2578]: https://tools.ietf.org/rfc/rfc2578.txt "RFC 2578"
[3413]: https://tools.ietf.org/rfc/rfc3413.txt "RFC 3413"
[3414]: https://tools.ietf.org/rfc/rfc3414.txt "RFC 3414"
[3416]: https://tools.ietf.org/rfc/rfc3416.txt "RFC 3416"
[3417]: https://tools.ietf.org/rfc/rfc3417.txt "RFC 3417"

[3826]: https://tools.ietf.org/rfc/rfc3826.txt "RFC 3826"

Constants

The following sections describe constants exported and used by this module.

snmp.Version1, snmp.Version2c, snmp.Version3

These constants are used to specify which of version supported by this module should be used.

snmp.ErrorStatus

This object contains constants for all valid values the error-status field in response PDUs can hold. If when parsing a PDU the error-index field contains a value not defined in this object the constant `snmp.ErrorStatus.GeneralError` will be used instead of the value in the error-status field. The following constants are defined in this object:

* `NoError`
* `TooBig`
* `NoSuchName`
* `BadValue`
* `ReadOnly`
* `GeneralError`
* `NoAccess`
* `WrongType`
* `WrongLength`
* `WrongEncoding`
* `WrongValue`
* `NoCreation`
* `InconsistentValue`
* `ResourceUnavailable`
* `CommitFailed`
* `UndoFailed`
* `AuthorizationError`
* `NotWritable`
* `InconsistentName`

snmp.ObjectType

This object contains constants used to specify syntax for varbind objects, e.g.:

```
var varbind = {
    oid: "1.3.6.1.2.1.1.4.0",
    type: snmp.ObjectType.OctetString,
    value: "user.name@domain.name"
```

};

The following constants are defined in this object:

* `Boolean`
* `Integer`
* `OctetString`
* `Null`
* `OID`
* `IpAddress`
* `Counter`
* `Gauge`
* `TimeTicks`
* `Opaque`
* `Integer32`
* `Counter32`
* `Gauge32`
* `Unsigned32`
* `Counter64`
* `NoSuchObject`
* `NoSuchInstance`
* `EndOfMibView`

snmp.TrapType

This object contains constants used to specify a type of SNMP trap. These constants are passed to the `trap()` and `inform()` methods exposed by the `Session` class. The following constants are defined in this object:

* `ColdStart`
* `WarmStart`
* `LinkDown`
* `LinkUp`
* `AuthenticationFailure`
* `EgpNeighborLoss`
* `EnterpriseSpecific`

snmp.PduType

This object contains constants used to identify the SNMP PDU types specified in RFC 3416. The values, along with their numeric codes, are:

* `160 - GetRequest`
* `161 - GetNextRequest`
* `162 - GetResponse`
* `163 - SetRequest`
* `164 - Trap`
* `165 - GetBulkRequest`

* `166 - InformRequest`
* `167 - TrapV2`
* `168 - Report`

snmp.SecurityLevel

This object contains constants to specify the security of an SNMPv3 message as per RFC 3414:
* `noAuthNoPriv` - for no message authentication or encryption
* `authNoPriv` - for message authentication and no encryption
* `authPriv` - for message authentication and encryption

snmp.AuthProtocols

This object contains constants to select a supported digest algorithm for SNMPv3 messages that require authentication:
* `md5` - for MD5 message authentication (HMAC-MD5-96)
* `sha` - for SHA message authentication (HMAC-SHA-96)

These are the two hash algorithms specified in RFC 3414. Other digest algorithms are not supported.

snmp.PrivProtocols

This object contains constants to select a supported encryption algorithm for SNMPv3 messages that require privacy:
* `des` - for DES encryption (CBC-DES)
* `aes` - for AES encryption (CFB-AES-128)

DES is the sole encryption algorithm specified in the original SNMPv3 User-Based Security Model RFC (RFC 3414); AES for SNMPv3 was added later in RFC 3826. Other encryption algorithms are not supported.

snmp.AgentXPduType

The Agent Extensibility (AgentX) Protocol specifies these PDUs in RFC 2741:
* `1 - Open`
* `2 - Close`
* `3 - Register`
* `4 - Unregister`
* `5 - Get`
* `6 - GetNext`
* `7 - GetBulk`
* `8 - TestSet`
* `9 - CommitSet`
* `10 - UndoSet`
* `11 - CleanupSet`
* `12 - Notify`

* `13 - Ping`
* `14 - IndexAllocate`
* `15 - IndexDeallocate`
* `16 - AddAgentCaps`
* `17 - RemoveAgentCaps`
* `18 - Response`

snmp.AccessControlModelType

* `None` - no access control for defined communities and users
* `Simple` - simple access control of levels "read-only" or "read-write" for defined communites and users

snmp.AccessLevel

- `None` - no access granted to the community or user
- `ReadOnly` - read-only access granted to the community or user
- `ReadWrite` - read-write access granted to the community or user

OID Strings & Varbinds

Some parts of this module accept simple OID strings, e.g.:

```
var oid = "1.3.6.1.2.1.1.5.0";
```

Other parts take an OID string, it's type and value. This is collectively referred to as a varbind, and is specified as an object, e.g.:

```
var varbind = {
    oid: "1.3.6.1.2.1.1.5.0",
    type: snmp.ObjectType.OctetString,
    value: new Buffer ("host1")
};
```

The `type` parameter is one of the constants defined in the `snmp.ObjectType` object.

The JavaScript `true` and `false` keywords are used for the values of varbinds with type `Boolean`.

All integer based types are specified as expected (this includes `Integer`, `Counter`, `Gauge`, `TimeTicks`, `Integer32`, `Counter32`, `Gauge32`, and `Unsigned32`), e.g. `-128` or `100`.

Since JavaScript does not offer full 64 bit integer support objects with type `Counter64` cannot be supported in the same way as other integer types, instead [Node.js][nodejs] `Buffer` objects are used. Users are responsible for producing (i.e. for `set()` requests) and consuming (i.e. the varbinds passed to callback functions) `Buffer` objects. That is, this module does not work with 64 bit integers, it simply treats them as opaque `Buffer` objects.

Dotted decimal strings are used for the values of varbinds with type `OID`, e.g. `1.3.6.1.2.1.1.5.0`.

Dotted quad formatted strings are used for the values of varbinds with type `IpAddress`, e.g. `192.168.1.1`.

[Node.js][nodejs] `Buffer` objects are used for the values of varbinds with type `Opaque` and `OctetString`. For varbinds with type `OctetString` this module will accept JavaScript strings, but will always give back `Buffer` objects.

The `NoSuchObject`, `NoSuchInstance` and `EndOfMibView` types are used to indicate an error condition. Currently there is no reason for users of this module to to build varbinds using these types.

[nodejs]: http://nodejs.org "Node.js"

Callback Functions & Error Handling

Most of the request methods exposed by this module require a mandatory callback function. This function is called once a request has been processed. This could be because an error occurred when processing the request, a trap has been dispatched or a successful response was received.

The first parameter to every callback is an error object. In the case no error occurred this parameter will be "null" indicating no error, e.g.:

```
function responseCb (error, varbinds) {
    if (error) {
        console.error (error);
    } else {
        // no error, do something with varbinds
    }
}
```

When defined, the error parameter is always an instance of the `Error` class, or a sub-class described in one of the sub-sections contained in this section.

The semantics of error handling is slightly different between SNMP version 1 and subsequent versions 2c and 3. In SNMP version 1 if an error occurs when calculating the value for one OID the request as a whole will fail, i.e. no OIDs will have a value.

This failure manifests itself within the error-status and error-index fields of the response. When the error-status field in the response is non-zero, i.e. not `snmp.ErrorStatus.NoError` the `callback` will be called with `error` defined detailing the error.

Requests made with SNMP version 1 can simply assume all OIDs have a value when no error object is passed to the `callback`, i.e.:

```
var oids = ["1.3.6.1.2.1.1.5.0", "1.3.6.1.2.1.1.6.0"];

session.get (oids, function (error, varbinds) {
    if (error) {
        console.error (error.toString ());
    } else {
        var sysName = varbinds[0].value; // this WILL have a value
    }
});
```

In SNMP versions 2c and 3, instead of using the error-status and error-index fields of the response to signal an error, the value for the varbind placed in the response for an OID will have an object syntax describing an error. The error-status and error-index fields of the response will indicate the request was successul, i.e. `snmp.ErrorStatus.NoError`.

This changes the way in which error checking is performed in the `callback`. When using SNMP version 2c each varbind must be checked to see if its value was computed and returned successfully:

```
var oids = ["1.3.6.1.2.1.1.5.0", "1.3.6.1.2.1.1.6.0"];

session.get (oids, function (error, varbinds) {
    if (error) {
        console.error (error.toString ());
    } else {
        if (varbinds[0].type != snmp.ErrorStatus.NoSuchObject
                && varbinds[0].type != snmp.ErrorStatus.NoSuchInstance
                && varbinds[0].type != snmp.ErrorStatus.EndOfMibView) {
            var sysName = varbinds[0].value;
        } else {
            console.error (snmp.ObjectType[varbinds[0].type] + ": "
                    + varbinds[0].oid);
        }
    }
});
```

This module exports two functions and promotes a specifc pattern to make error checking a little simpler. Firstly, regardless of version in use varbinds can always be checked. This results in a generic `callback` that can be used for both versions.

The `isVarbindError()` function can be used to determine if a varbind has an error condition. This function takes a single `varbind` parameter and returns `true` if the varbind has an error condition, otherwise `false`. The exported `varbindError()` function can then be used to obtain the error string describing the error, which will include the OID for the varbind:

```
  session.get (oids, function (error, varbinds) {
      if (error) {
          console.error (error.toString ());
      } else {
          if (snmp.isVarbindError (varbinds[0])) {
              console.error (snmp.varbindError (varbinds[0]));
          } else {
              var sysName = varbinds[0].value;
          }
      }
  });
```

If the `varbindError` function is called with a varbind for which `isVarbindError` would return false, the string `NotAnError` will be returned appended with the related OID.

The sections following defines the error classes used by this module.

snmp.RequestFailedError

This error indicates a remote host failed to process a request. The exposed `message` attribute will contain a detailed error message. This error also exposes a `status` attribute which contains the error-index value from a response. This will be one of the constants defined in the `snmp.ErrorStatus` object.

snmp.RequestInvalidError

This error indicates a failure to render a request message before it could be sent. The error can also indicate that a parameter provided was invalid. The exposed `message` attribute will contain a detailed error message.

snmp.RequestTimedOutError

This error states that no response was received for a particular request. The exposed `message` attribute will contain the value `Request timed out`.

snmp.ResponseInvalidError

This error indicates a failure to parse a response message. The exposed `message` attribute will contain a detailed error message.

Using This Module: Command & Notification Generator

This library provides a `Session` class to provide support for building "Command Generator" and "Notification Originator" SNMP applications.

All SNMP requests are made using an instance of the `Session` class. This module exports two functions that are used to create instances of the `Session` class:

* `createSession()` - for v1 and v2c sessions
* `createV3Session()` - for v3 sessions

snmp.createSession ([target], [community], [options])

The `createSession()` function instantiates and returns an instance of the `Session` class for SNMPv1 or SNMPv2c:

```
// Default options
var options = {
    port: 161,
    retries: 1,
    timeout: 5000,
    backoff: 1.0,
    transport: "udp4",
    trapPort: 162,
    version: snmp.Version1,
    backwardsGetNexts: true,
    idBitsSize: 32
};

var session = snmp.createSession ("127.0.0.1", "public", options);
```

The optional `target` parameter defaults to `127.0.0.1`. The optional `community` parameter defaults to `public`. The optional `options` parameter is an object, and can contain the following items:

* `port` - UDP port to send requests too, defaults to `161`
* `retries` - Number of times to re-send a request, defaults to `1`
* `sourceAddress` - IP address from which SNMP requests should originate, there is no default for this option, the operating system will select an appropriate source address when the SNMP request is sent
* `sourcePort` - UDP port from which SNMP requests should originate, defaults to an ephemeral port selected by the operation system
* `timeout` - Number of milliseconds to wait for a response before re-trying or failing, defaults to `5000`
* `backoff` - The factor by which to increase the `timeout` for every retry, defaults to `1` for no increase
* `transport` - Specify the transport to use, can be either `udp4` or `udp6`, defaults to `udp4`

* `trapPort` - UDP port to send traps and informs too, defaults to `162`
* `version` - Either `snmp.Version1` or `snmp.Version2c`, defaults to
  `snmp.Version1`
* `backwardsGetNexts` - boolean to allow GetNext operations to retrieve lexicographically
  preceeding OIDs
* `idBitsSize` - Either `16` or `32`, defaults to `32`. Used to reduce the size
  of the generated id for compatibility with some older devices.

When a session has been finished with it should be closed:

```
session.close ();
```

snmp.createV3Session (target, user, [options])

The `createV3Session()` function instantiates and returns an instance of the
same `Session` class as `createSession()`, only instead initialized for SNMPv3:

```
// Default options for v3
var options = {
    port: 161,
    retries: 1,
    timeout: 5000,
    transport: "udp4",
    trapPort: 162,
    version: snmp.Version3,
    idBitsSize: 32,
    context: ""
};

// Example user
var user = {
    name: "blinkybill",
    level: snmp.SecurityLevel.authPriv,
    authProtocol: snmp.AuthProtocols.sha,
    authKey: "madeahash",
    privProtocol: snmp.PrivProtocols.des,
    privKey: "privycouncil"
};

var session = snmp.createV3Session ("127.0.0.1", user, options);
```

The `target` and `user` parameters are mandatory. The optional `options` parameter
has the same meaning as for the `createSession()` call. The one additional field
in the options parameter is the `context` field, which adds an SNMPv3 context to
the session.

The `user` object must contain a `name` and `level` field. The `level` field can
take these values from the `snmp.SecurityLevel` object:

* `snmp.SecurityLevel.noAuthNoPriv` - for no message authentication or encryption
* `snmp.SecurityLevel.authNoPriv` - for message authentication and no encryption
* `snmp.SecurityLevel.authPriv` - for message authentication and encryption

The meaning of these are as per RFC3414. If the `level` supplied is `authNoPriv` or `authPriv`, then the `authProtocol` and `authKey` fields must also be present. The `authProtocol` field can take values from the `snmp.AuthProtocols` object:
* `snmp.AuthProtocols.md5` - for MD5 message authentication
* `snmp.AuthProtocols.sha` - for SHA message authentication

If the `level` supplied is `authPriv`, then the `privProtocol` and `privKey` fields must also be present. The `privProtocol` field can take values from the `snmp.PrivProtocols` object:
* `snmp.PrivProtocols.des` - for DES encryption
* `snmp.PrivProtocols.aes` - for AES encryption

Once a v3 session is created, the same set of `session` methods are available as for v1 and v2c.

session.on ("close", callback)

The `close` event is emitted by the session when the sessions underlying UDP socket is closed.

No arguments are passed to the callback.

Before this event is emitted all outstanding requests are cancelled, resulting in the failure of each outstanding request. The error passed back through to each request will be an instance of the `Error` class with the errors `message` attribute set to `Socket forcibly closed`.

The following example prints a message to the console when a sessions underlying UDP socket is closed:

```
session.on ("close", function () {
    console.log ("socket closed");
});
```

session.on ("error", callback)

The `error` event is emitted by the session when the sessions underlying UDP socket emits an error.

The following arguments will be passed to the `callback` function:

* `error` - An instance of the `Error` class, the exposed `message` attribute will contain a detailed error message.

The following example prints a message to the console when an error occurs with a sessions underlying UDP socket, the session is then closed:

```
session.on ("error", function (error) {
    console.log (error.toString ());
    session.close ();
});
```

session.close ()

The `close()` method closes the sessions underlying UDP socket. This will result in the `close` event being emitted by the sessions underlying UDP socket which is passed through to the session, resulting in the session also emitting a `close` event.

The following example closes a sessions underlying UDP socket:

```
session.close ();
```

session.get (oids, callback)

The `get()` method fetches the value for one or more OIDs.

The `oids` parameter is an array of OID strings. The `callback` function is called once the request is complete. The following arguments will be passed to the `callback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no error occurred
* `varbinds` - Array of varbinds, will not be provided if an error occurred

The varbind in position N in the `varbinds` array will correspond to the OID in position N in the `oids` array in the request.

Each varbind must be checked for an error condition using the `snmp.isVarbindError()` function when using SNMP version 2c.

The following example fetches values for the sysName (`1.3.6.1.2.1.1.5.0`) and sysLocation (`1.3.6.1.2.1.1.6.0`) OIDs:

```
var oids = ["1.3.6.1.2.1.1.5.0", "1.3.6.1.2.1.1.6.0"];

session.get (oids, function (error, varbinds) {
    if (error) {
        console.error (error.toString ());
    } else {
        for (var i = 0; i < varbinds.length; i++) {
            // for version 1 we can assume all OIDs were successful
```

```
            console.log (varbinds[i].oid + "|" + varbinds[i].value);

            // for version 2c we must check each OID for an error condition
            if (snmp.isVarbindError (varbinds[i]))
                console.error (snmp.varbindError (varbinds[i]));
            else
                console.log (varbinds[i].oid + "|" + varbinds[i].value);
        }
    }
});
```

session.getBulk (oids, [nonRepeaters], [maxRepetitions], callback)

The `getBulk()` method fetches the value for the OIDs lexicographically following one or more OIDs in the MIB tree.

The `oids` parameter is an array of OID strings. The optional `nonRepeaters` parameter specifies the number of OIDs in the `oids` parameter for which only 1 varbind should be returned, and defaults to `0`. For each remaining OID in the `oids` parameter the optional `maxRepetitions` parameter specifies how many OIDs lexicographically following an OID for which varbinds should be fetched, and defaults to `20`.

The `callback` function is called once the request is complete. The following arguments will be passed to the `callback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no
  error occurred
* `varbinds` - Array of varbinds, will not be provided if an error occurred

The varbind in position N in the `varbinds` array will correspond to the OID in position N in the `oids` array in the request.

For for the first `nonRepeaters` items in `varbinds` each item will be a single varbind. For all remaining items in `varbinds` each item will be an array of varbinds - this makes it easy to tie response varbinds with requested OIDs since response varbinds are grouped and placed in the same position in `varbinds`.

Each varbind must be checked for an error condition using the `snmp.isVarbindError()` function when using SNMP version 2c.

The following example fetches values for the OIDs following the sysContact (`1.3.6.1.2.1.1.4.0`) and sysName (`1.3.6.1.2.1.1.5.0`) OIDs, and up to the first 20 OIDs in the ifDescr (`1.3.6.1.2.1.2.2.1.2`) and ifType (`1.3.6.1.2.1.2.2.1.3`) columns from the ifTable (`1.3.6.1.2.1.2.2`) table:

```
var oids = [
```

```
        "1.3.6.1.2.1.1.4.0",
        "1.3.6.1.2.1.1.5.0",
        "1.3.6.1.2.1.2.2.1.2",
        "1.3.6.1.2.1.2.2.1.3"
    ];

    var nonRepeaters = 2;

    session.getNext (oids, nonRepeaters, function (error, varbinds) {
        if (error) {
            console.error (error.toString ());
        } else {
            // step through the non-repeaters which are single varbinds
            for (var i = 0; i < nonRepeaters; i++) {
                if (i >= varbinds.length)
                    break;

                if (snmp.isVarbindError (varbinds[i]))
                    console.error (snmp.varbindError (varbinds[i]));
                else
                    console.log (varbinds[i].oid + "|" + varbinds[i].value);
            }

            // then step through the repeaters which are varbind arrays
            for (var i = nonRepeaters; i < varbinds.length; i++) {
                for (var j = 0; j < varbinds[i].length; j++) {
                    if (snmp.isVarbindError (varbinds[i][j]))
                        console.error (snmp.varbindError (varbinds[i][j]));
                    else
                        console.log (varbinds[i][j].oid + "|"
                          + varbinds[i][j].value);
                }
        }
    });
```

session.getNext (oids, callback)

The `getNext()` method fetches the value for the OIDs lexicographically
following one or more OIDs in the MIB tree.

The `oids` parameter is an array of OID strings. The `callback` function is
called once the request is complete. The following arguments will be passed
to the `callback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no
  error occurred
* `varbinds` - Array of varbinds, will not be provided if an error occurred

The varbind in position N in the `varbinds` array will correspond to the OID in position N in the `oids` array in the request.

Each varbind must be checked for an error condition using the `snmp.isVarbindError()` function when using SNMP version 2c.

The following example fetches values for the next OIDs following the sysObjectID (`1.3.6.1.2.1.1.1.0`) and sysName (`1.3.6.1.2.1.1.4.0`) OIDs:

```
var oids = [
    "1.3.6.1.2.1.1.1.0",
    "1.3.6.1.2.1.1.4.0"
];

session.getNext (oids, function (error, varbinds) {
    if (error) {
        console.error (error.toString ());
    } else {
        for (var i = 0; i < varbinds.length; i++) {
            // for version 1 we can assume all OIDs were successful
            console.log (varbinds[i].oid + "|" + varbinds[i].value);

            // for version 2c we must check each OID for an error condition
            if (snmp.isVarbindError (varbinds[i]))
                console.error (snmp.varbindError (varbinds[i]));
            else
                console.log (varbinds[i].oid + "|" + varbinds[i].value);
        }
    }
});
```

session.inform (typeOrOid, [varbinds], [options], callback)

The `inform()` method sends a SNMP inform.

The `typeOrOid` parameter can be one of two types; one of the constants defined in the `snmp.TrapType` object (excluding the `snmp.TrapType.EnterpriseSpecific` constant), or an OID string.

The first varbind to be placed in the request message will be for the `sysUptime.0` OID (`1.3.6.1.2.1.1.3.0`). The value for this varbind will be the value returned by the `process.uptime ()` function multiplied by 100 (this can be overridden by providing `upTime` in the optional `options` parameter, as documented below).

This will be followed by a second varbind for the `snmpTrapOID.0` OID (`1.3.6.1.6.3.1.1.4.1.0`). The value for this will depend on the `typeOrOid` parameter. If a constant is specified the trap OID for the constant will be used as supplied for the varbinds value, otherwise the OID string specified will be used as is for the value of the varbind.

The optional `varbinds` parameter is an array of varbinds to include in the inform request, and defaults to the empty array `[]`.

The optional `options` parameter is an object, and can contain the following items:

* `upTime` - Value of the `sysUptime.0` OID (`1.3.6.1.2.1.1.3.0`) in the inform, defaults to the value returned by the `process.uptime ()` function multiplied by 100

The `callback` function is called once a response to the inform request has been received, or an error occurred. The following arguments will be passed to the `callback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no error occurred
* `varbinds` - Array of varbinds, will not be provided if an error occurred

The varbind in position N in the `varbinds` array will correspond to the varbind in position N in the `varbinds` array in the request. The remote host should echo back varbinds and their values as specified in the request, and the `varbinds` array will contain each varbind as sent back by the remote host.

Normally there is no reason to use the contents of the `varbinds` parameter since the varbinds are as they were sent in the request.

The following example sends a generic cold-start inform to a remote host, it does not include any varbinds:

```
session.inform (snmp.TrapType.ColdStart, function (error) {
    if (error)
        console.error (error);
});
```

The following example sends an enterprise specific inform to a remote host, and includes two enterprise specific varbinds:

```
var informOid = "1.3.6.1.4.1.2000.1";

var varbinds = [
    {
        oid: "1.3.6.1.4.1.2000.2",
        type: snmp.ObjectType.OctetString,
        value: "Periodic hardware self-check"
    },
```

```
    {
        oid: "1.3.6.1.4.1.2000.3",
        type: snmp.ObjectType.OctetString,
        value: "hardware-ok"
    }
];

// Override sysUpTime, specfiying it as 10 seconds...
var options = {upTime: 1000};
session.inform (informOid, varbinds, options, function (error) {
    if (error)
        console.error (error);
});
```

session.set (varbinds, callback)

The `set()` method sets the value of one or more OIDs.

The `varbinds` parameter is an array of varbind objects. The `callback` function is called once the request is complete. The following arguments will be passed to the `callback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no error occurred
* `varbinds` - Array of varbinds, will not be provided if an error occurred

The varbind in position N in the `varbinds` array will correspond to the varbind in position N in the `varbinds` array in the request. The remote host should echo back varbinds and their values as specified in the request unless an error occurred. The `varbinds` array will contain each varbind as sent back by the remote host.

Each varbind must be checked for an error condition using the `snmp.isVarbindError()` function when using SNMP version 2c.

The following example sets the value of the sysName (`1.3.6.1.2.1.1.4.0`) and sysLocation (`1.3.6.1.2.1.1.6.0`) OIDs:

```
var varbinds = [
    {
        oid: "1.3.6.1.2.1.1.5.0",
        type: snmp.ObjectType.OctetString,
        value: "host1"
    }, {
        oid: "1.3.6.1.2.1.1.6.0",
        type: snmp.ObjectType.OctetString,
        value: "somewhere"
    }
```

```
    ];

    session.set (varbinds, function (error, varbinds) {
        if (error) {
            console.error (error.toString ());
        } else {
            for (var i = 0; i < varbinds.length; i++) {
                // for version 1 we can assume all OIDs were successful
                console.log (varbinds[i].oid + "|" + varbinds[i].value);

                // for version 2c we must check each OID for an error condition
                if (snmp.isVarbindError (varbinds[i]))
                    console.error (snmp.varbindError (varbinds[i]));
                else
                    console.log (varbinds[i].oid + "|" + varbinds[i].value);
            }
        }
    });
```

session.subtree (oid, [maxRepetitions], feedCallback, doneCallback)

The `subtree()` method fetches the value for all OIDs lexicographically following a specified OID in the MIB tree which have the specified OID as their base. For example, the OIDs sysName (`1.3.6.1.2.1.1.5.0`) and sysLocation (`1.3.6.1.2.1.1.6.0`) both have the same base system (`1.3.6.1.2.1.1`) OID.

For SNMP version 1 repeated `get()` calls are made until the one of the returned OIDs does not use the specified OID as its base. For SNMP version 2c repeated `getBulk()` calls are made until the one of the returned OIDs does no used the specified OID as its base.

The `oid` parameter is an OID string. The optional `maxRepetitions` parameter is passed to `getBulk()` requests when SNMP version 2c is being used.

This method will not call a single callback once all OID values are fetched. Instead the `feedCallback` function will be called each time a response is received from the remote host. The following arguments will be passed to the `feedCallback` function:

* `varbinds` - Array of varbinds, and will contain at least one varbind

Each varbind must be checked for an error condition using the `snmp.isVarbindError()` function when using SNMP version 2c.

Once at least one of the returned OIDs does not use the specified OID as its base, or an error has occurred, the `doneCallback` function will be called. The following arguments will be passed to the `doneCallback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no
  error occurred

Once the `doneCallback` function has been called the request is complete and the `feedCallback` function will no longer be called.

If the `feedCallback` function returns a `true` value when called no more `get()` or `getBulk()` method calls will be made and the `doneCallback` will be called.

The following example fetches all OIDS under the system (`1.3.6.1.2.1.1`) OID:

```
var oid = "1.3.6.1.2.1.1";

function doneCb (error) {
    if (error)
        console.error (error.toString ());
}

function feedCb (varbinds) {
    for (var i = 0; i < varbinds.length; i++) {
        if (snmp.isVarbindError (varbinds[i]))
            console.error (snmp.varbindError (varbinds[i]));
        else
            console.log (varbinds[i].oid + "|" + varbinds[i].value);
    }
}

var maxRepetitions = 20;

// The maxRepetitions argument is optional, and will be ignored unless using
// SNMP verison 2c
session.subtree (oid, maxRepetitions, feedCb, doneCb);
```

session.table (oid, [maxRepetitions], callback)

The `table()` method fetches the value for all OIDs lexicographically following a specified OID in the MIB tree which have the specified OID as their base, much like the `subtree()` method.

This method is designed to fetch conceptial tables, for example the ifTable (`1.3.6.1.2.1.2.2`) table. The values for returned varbinds will be structured into objects to represent conceptual rows. Each row is then placed into an object with the rows index being the key, e.g.:

```
var table = {
    // Rows keyed by ifIndex (1 and 2 are shown)
```

```
    1: {
        // ifDescr (column 2) and ifType (columnd 3) are shown
        2: "interface-1",
        3: 6,
        ...
    },
    2: {
        2: "interface-2",
        3: 6,
        ...
    },
    ...
}
```

Internally this method calls the `subtree()` method to obtain the subtree of the specified table.

The `oid` parameter is an OID string. If an OID string is passed which does not represent a table the resulting object produced to hold table data will be empty, i.e. it will contain no indexes and rows. The optional `maxRepetitions` parameter is passed to the `subtree()` request.

The `callback` function will be called once the entire table has been fetched. The following arguments will be passed to the `callback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no error occurred
* `table` - Object containing object references representing conceptual rows keyed by index (e.g. for the ifTable table rows are keyed by ifIndex), each row object will contain values keyed by column number, will not be provided if an error occurred

If an error occurs with any varbind returned by `subtree()` no table will be passed to the `callback` function. The reason for failure, and the related OID string (as returned from a call to the `snmp.varbindError()` function), will be passed to the `callback` function in the `error` argument as an instance of the `RequestFailedError` class.

The following example fetches the ifTable (`1.3.6.1.2.1.2.2`) table:

```
var oid = "1.3.6.1.2.1.2.2";

function sortInt (a, b) {
    if (a > b)
        return 1;
    else if (b > a)
        return -1;
    else
```

```
        return 0;
    }

    function responseCb (error, table) {
        if (error) {
            console.error (error.toString ());
        } else {
            // This code is purely used to print rows out in index order,
            // ifIndex's are integers so we'll sort them numerically using
            // the sortInt() function above
            var indexes = [];
            for (index in table)
                indexes.push (parseInt (index));
            indexes.sort (sortInt);

            // Use the sorted indexes we've calculated to walk through each
            // row in order
            for (var i = 0; i < indexes.length; i++) {
                // Like indexes we sort by column, so use the same trick here,
                // some rows may not have the same columns as other rows, so
                // we calculate this per row
                var columns = [];
                for (column in table[indexes[i]])
                    columns.push (parseInt (column));
                columns.sort (sortInt);

                // Print index, then each column indented under the index
                console.log ("row for index = " + indexes[i]);
                for (var j = 0; j < columns.length; j++) {
                    console.log ("    column " + columns[j] + " = "
                            + table[indexes[i]][columns[j]]);
                }
            }
        }
    }

    var maxRepetitions = 20;

    // The maxRepetitions argument is optional, and will be ignored unless using
    // SNMP verison 2c
    session.table (oid, maxRepetitions, responseCb);
```

session.tableColumns (oid, columns, [maxRepetitions], callback)

The `tableColumns()` method implements the same interface as the `table()` method. However, only the columns specified in the `columns` parameter will be in the resulting table.

This method should be used when only selected columns are required, and will be many times faster than the `table()` method since a much smaller amount of data will be fected.

The following example fetches the ifTable (`1.3.6.1.2.1.2.2`) table, and specifies that only the ifDescr (`1.3.6.1.2.1.2.2.1.2`) and ifPhysAddress (`1.3.6.1.2.1.2.2.1.6`) columns should actually be fetched:

```
var oid = "1.3.6.1.2.1.2.2";
var columns = [2, 6];

function sortInt (a, b) {
    if (a > b)
        return 1;
    else if (b > a)
        return -1;
    else
        return 0;
}

function responseCb (error, table) {
    if (error) {
        console.error (error.toString ());
    } else {
        // This code is purely used to print rows out in index order,
        // ifIndex's are integers so we'll sort them numerically using
        // the sortInt() function above
        var indexes = [];
        for (index in table)
            indexes.push (parseInt (index));
        indexes.sort (sortInt);

        // Use the sorted indexes we've calculated to walk through each
        // row in order
        for (var i = 0; i < indexes.length; i++) {
            // Like indexes we sort by column, so use the same trick here,
            // some rows may not have the same columns as other rows, so
            // we calculate this per row
            var columns = [];
            for (column in table[indexes[i]])
                columns.push (parseInt (column));
            columns.sort (sortInt);

            // Print index, then each column indented under the index
            console.log ("row for index = " + indexes[i]);
            for (var j = 0; j < columns.length; j++) {
                console.log ("   column " + columns[j] + " = "
                        + table[indexes[i]][columns[j]]);
```

```
            }
        }
    }
  }

  var maxRepetitions = 20;

  // The maxRepetitions argument is optional, and will be ignored unless using
  // SNMP verison 2c
  session.tableColumns (oid, columns, maxRepetitions, responseCb);
```

session.trap (typeOrOid, [varbinds], [agentAddrOrOptions], callback)

The `trap()` method sends a SNMP trap.

The `typeOrOid` parameter can be one of two types; one of the constants defined in the `snmp.TrapType` object (excluding the `snmp.TrapType.EnterpriseSpecific` constant), or an OID string.

For SNMP version 1 when a constant is specified the following fields are set in the trap:

* The enterprise field is set to the OID `1.3.6.1.4.1`
* The generic-trap field is set to the constant specified
* The specific-trap field is set to 0

When an OID string is specified the following fields are set in the trap:

* The final decimal is stripped from the OID string and set in the specific-trap field
* The remaining OID string is set in the enterprise field
* The generic-trap field is set to the constant `snmp.TrapType.EnterpriseSpecific`

In both cases the time-stamp field in the trap PDU is set to the value returned by the `process.uptime ()` function multiplied by `100`.

SNMP version 2c messages are quite different in comparison with version 1. The version 2c trap has a much simpler format, simply a sequence of varbinds. The first varbind to be placed in the trap message will be for the `sysUptime.0` OID (`1.3.6.1.6.3.1.1.4.1.0`). The value for this varbind will be the value returned by the `process.uptime ()` function multiplied by 100 (this can be overridden by providing `upTime` in the optional `options` parameter, as documented below).

This will be followed by a second varbind for the `snmpTrapOID.0` OID (`1.3.6.1.6.3.1.1.4.1.0`). The value for this will depend on the `typeOrOid` parameter. If a constant is specified the trap OID for the constant will be used as supplied for the varbinds value, otherwise the OID string specified will be used as is for the value of the varbind.

The optional `varbinds` parameter is an array of varbinds to include in the trap, and defaults to the empty array `[]`.

The optional `agentAddrOrOptions` parameter can be one of two types; one is the IP address used to populate the agent-addr field for SNMP version 1 type traps, and defaults to `127.0.0.1`, or an object, and can contain the following items:

* `agentAddr` - IP address used to populate the agent-addr field for SNMP version 1 type traps, and defaults to `127.0.0.1`
* `upTime` - Value of the `sysUptime.0` OID (`1.3.6.1.6.3.1.1.4.1.0`) in the trap, defaults to the value returned by the `process.uptime ()` function multiplied by 100

NOTE When using SNMP version 2c the `agentAddr` parameter is ignored if specified since version 2c trap messages do not have an agent-addr field.

The `callback` function is called once the trap has been sent, or an error occurred. The following arguments will be passed to the `callback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no error occurred

The following example sends an enterprise specific trap to a remote host using a SNMP version 1 trap, and includes the sysName (`1.3.6.1.2.1.1.5.0`) varbind in the trap. Before the trap is sent the `agentAddr` field is calculated using DNS to resolve the hostname of the local host:

```
var enterpriseOid = "1.3.6.1.4.1.2000.1"; // made up, but it may be valid

var varbinds = [
    {
        oid: "1.3.6.1.2.1.1.5.0",
        type: snmp.ObjectType.OctetString,
        value: "host1"
    }
];

dns.lookup (os.hostname (), function (error, agentAddress) {
    if (error) {
        console.error (error);
    } else {
        // Override sysUpTime, specfiying it as 10 seconds...
        var options = {agentAddr: agentAddress, upTime: 1000};
        session.trap (enterpriseOid, varbinds, agentAddress,
```

```
                function (error) {
            if (error)
                console.error (error);
        });
    }
});
```

The following example sends a generic link-down trap to a remote host using a SNMP version 1 trap, it does not include any varbinds or specify the `agentAddr` parameter:

```
session.trap (snmp.TrapType.LinkDown, function (error) {
    if (error)
        console.error (error);
});
```

The following example sends an enterprise specific trap to a remote host using a SNMP version 2c trap, and includes two enterprise specific varbinds:

```
var trapOid = "1.3.6.1.4.1.2000.1";

var varbinds = [
    {
        oid: "1.3.6.1.4.1.2000.2",
        type: snmp.ObjectType.OctetString,
        value: "Hardware health status changed"
    },
    {
        oid: "1.3.6.1.4.1.2000.3",
        type: snmp.ObjectType.OctetString,
        value: "status-error"
    }
];

// version 2c should have been specified when creating the session
session.trap (trapOid, varbinds, function (error) {
    if (error)
        console.error (error);
});
```

session.walk (oid, [maxRepetitions], feedCallback, doneCallback)

The `walk()` method fetches the value for all OIDs lexicographically following a specified OID in the MIB tree.

For SNMP version 1 repeated `get()` calls are made until the end of the MIB tree is reached. For SNMP version 2c repeated `getBulk()` calls are made until the end of the MIB tree is reached.

The `oid` parameter is an OID string. The optional `maxRepetitions` parameter is passed to `getBulk()` requests when SNMP version 2c is being used.

This method will not call a single callback once all OID values are fetched. Instead the `feedCallback` function will be called each time a response is received from the remote host. The following arguments will be passed to the `feedCallback` function:

* `varbinds` - Array of varbinds, and will contain at least one varbind

Each varbind must be checked for an error condition using the `snmp.isVarbindError()` function when using SNMP version 2c.

Once the end of the MIB tree has been reached, or an error has occurred, the `doneCallback` function will be called. The following arguments will be passed to the `doneCallback` function:

* `error` - Instance of the `Error` class or a sub-class, or `null` if no error occurred

Once the `doneCallback` function has been called the request is complete and the `feedCallback` function will no longer be called.

If the `feedCallback` function returns a `true` value when called no more `get()` or `getBulk()` method calls will be made and the `doneCallback` will be called.

The following example walks to the end of the MIB tree starting from the ifTable (`1.3.6.1.2.1.2.2`) OID:

```
var oid = "1.3.6.1.2.1.2.2";

function doneCb (error) {
    if (error)
        console.error (error.toString ());
}

function feedCb (varbinds) {
    for (var i = 0; i < varbinds.length; i++) {
        if (snmp.isVarbindError (varbinds[i]))
            console.error (snmp.varbindError (varbinds[i]));
        else
            console.log (varbinds[i].oid + "|" + varbinds[i].value);
    }
}

var maxRepetitions = 20;
```

```
// The maxRepetitions argument is optional, and will be ignored unless using
// SNMP verison 2c
session.walk (oid, maxRepetitions, feedCb, doneCb);
```

Using This Module: Notification Receiver

RFC 3413 classifies a "Notification Receiver" SNMP application that receives "Notification-Class" PDUs. Notifications include both SNMP traps and informs. This library is able to receive all types of notification PDU:

* `Trap-PDU` (original v1 trap PDUs, which are now considered obselete)
* `Trapv2-PDU` (unacknowledged notifications)
* `InformRequest-PDU` (same format as `Trapv2-PDU` but with message acknowledgement)

The library provides a `Receiver` class for receiving SNMP notifications. This module exports the `createReceiver()` function, which creates a new `Receiver` instance.

The receiver creates an `Authorizer` instance to control incoming access. More detail on this is found below in the [Authorizer Module](#authorizer-module) section below.

snmp.createReceiver (options, callback)

The `createReceiver()` function instantiates and returns an instance of the `Receiver` class:

```
// Default options
var options = {
    port: 162,
    disableAuthorization: false,
    accessControlModelType: snmp.AccessControlModelType.None,
    engineID: "8000B98380XXXXXXXXXXXX", // where the X's are random hex digits
    transport: "udp4"
};

var callback = function (error, notification) {
    if ( error ) {
        console.error (error);
    } else {
        console.log (JSON.stringify(notification, null, 2));
    }
};

receiver = snmp.createReceiver (options, callback);
```

The `options` and `callback` parameters are mandatory. The `options` parameter is an object, possibly empty, and can contain the following fields:

* `port` - the port to listen for notifications on - defaults to 162. Note that binding to port 162 on some systems requires the receiver process to be run with administrative privilege. If this is not possible then choose a port greater than 1024.
* `disableAuthorization` - disables local authorization for all community-based notifications received and for those user-based notifications received with no message authentication or privacy (noAuthNoPriv) - defaults to false
* `engineID` - the engineID used for SNMPv3 communications, given as a hex string - defaults to a system-generated engineID containing elements of random
* `transport` - the transport family to use - defaults to `udp4`

The `callback` parameter is a callback function of the form `function (error, notification)`. On an error condition, the `notification` parameter is set to `null`. On successful reception of a notification, the error parameter is set to `null`, and the `notification` parameter is set as an object with the notification PDU details in the `pdu` field and the sender socket details in the `rinfo` field. For example:

```
{
    "pdu": {
        "type": 166,
        "id": 45385686,
        "varbinds": [
            {
                "oid": "1.3.6.1.2.1.1.3.0",
                "type": 67,
                "value": 5
            },
            {
                "oid": "1.3.6.1.6.3.1.1.4.1.0",
                "type": 6,
                "value": "1.3.6.1.6.3.1.1.5.2"
            }
        ],
        "scoped": false
    },
    "rinfo": {
        "address": "127.0.0.1",
        "family": "IPv4",
        "port": 43162,
        "size": 72
    }
}
```

receiver.getAuthorizer ()

Returns the receiver's `Authorizer` instance, used to control access to the receiver. See the `Authorizer` section for further details.

receiver.close ()

Closes the receiver's listening socket, ending the operation of the receiver.

Using This Module: SNMP Agent

The SNMP agent responds to all four "request class" PDUs relevant to a Command Responder application:

* **GetRequest** - request exactly matched OID instances
* **GetNextRequest** - request lexicographically "next" OID instances in the MIB tree
* **GetBulkRequest** - request a series of "next" OID instances in the MIB tree
* **SetRequest** - set values for specified OIDs

The agent sends a **GetResponse** PDU to all four request PDU types, in conformance to RFC 3416.

The agent - like the notification receiver - maintains an `Authorizer` instance
to control access to the agent, details of which are in the [Authorizer Module](#authorizer-module)
section below.

The central data structure that the agent maintains is a `Mib` instance, the API of which is
detailed in the [Mib Module](#mib-module) section below. The agent allows the MIB to be queried
and manipulated through the API, as well as queried and manipulated through the SNMP interface with
the above four request-class PDUs.

The agent also supports SNMP proxy forwarder applications with its singleton `Forwarder` instance,
which is documented in the [Forwarder Module](#forwarder-module) section below.

snmp.createAgent (options, callback, mib)

The `createAgent()` function instantiates and returns an instance of the `Agent`
class:

```
// Default options
var options = {
    port: 161,
    disableAuthorization: false,
    accessControlModelType: snmp.AccessControlModelType.None,
    engineID: "8000B98380XXXXXXXXXXXX", // where the X's are random hex digits
    transport: "udp4"
};

var callback = function (error, data) {
    if ( error ) {
        console.error (error);
    } else {
```

```
        console.log (JSON.stringify(data, null, 2));
    }
};

agent = snmp.createAgent (options, callback);
```

The `options` and `callback` parameters are mandatory. The `options` parameter is an object, possibly empty, and can contain the following fields:

* `port` - the port for the agent to listen on - defaults to 161. Note that binding to port 161 on some systems requires the receiver process to be run with administrative privilege. If this is not possible, then choose a port greater than 1024.
* `disableAuthorization` - disables local authorization for all community-based notifications received and for those user-based notifications received with no message authentication or privacy (noAuthNoPriv) - defaults to false
* `accessControlModelType` - specifies which access control model to use. Defaults to `snmp.AccessControlModelType.None`, but can be set to `snmp.AccessControlModelType.Simple` for further access control capabilities. See the `Authorization` class description for more information.
* `engineID` - the engineID used for SNMPv3 communications, given as a hex string - defaults to a system-generated engineID containing elements of random
* `transport` - the transport family to use - defaults to `udp4`

The `mib` parameter is optional, and sets the agent's singleton `Mib` instance. If not supplied, the agent creates itself a new empty `Mib` singleton. If supplied, the `Mib` instance needs to be created and populated as per the [Mib Module](#mib-module) section below.

agent.getAuthorizer ()

Returns the agent's singleton `Authorizer` instance, used to control access to the agent. See the `Authorizer` section for further details.

agent.getMib ()

Returns the agent's singleton `Mib` instance, which holds all of the management data for the agent.

agent.setMib (mib)

Sets the agent's singleton `Mib` instance to the supplied one. The agent discards its existing `Mib` instance.

agent.getForwarder ()

Returns the agent's singleton `Forwarder` instance, which holds a list of registered proxies that specify context-based forwarding to remote hosts.

agent.close ()

Closes the agent's listening socket, ending the operation of the agent.

Authorizer Module

Both the receiver and agent maintain an singleton `Authorizer` instance, which is responsible for maintaining an authorization list of SNMP communities (for v1 and v2c notifications) and also an authorization list of SNMP users (for v3 notifications). These lists are used to authorize notification access to the receiver, and to store security protocol and key settings. RFC 3414 terms the user list as the the "usmUserTable" stored in the receiver's "Local Configuration Database".

If a v1 or v2c notification is received with a community that is not in the receiver's community authorization list, the receiver will not accept the notification, instead returning a error of class `RequestFailedError` to the supplied callback function. Similarly, if a v3 notification is received with a user whose name is not in the receiver's user authorization list, the receiver will return a `RequestFailedError`. If the `disableAuthorization` option is supplied for the receiver on start-up, then these local authorization list checks are disabled for community notifications and noAuthNoPriv user notifications. Note that even with this setting, the user list is *still checked* for authNoPriv and authPriv notifications, as the library still requires access to the correct keys for the message authentication and encryption operations, and these keys are stored against a user in the user authorization list.

The API allows the receiver's / agent's community authorization and user authorization lists to be managed with adds, queries and deletes.

For an agent, there is a further optional access control check, that can limit the access for a given community or user according to the `AccessControlModelType` supplied as an option to the agent. The default model type is `snmp.AccessControlModelType.None`, which means that - after the authorization list checks described in the preceding paragraphs - there is no further access control restrictions i.e. all requests are granted access by the agent. A second access control model type `snmp.AccessControlModelType.Simple` can be selected, which creates a `SimpleAccessControlModel` object that can be manipulated to specify that a community or user has one of three levels of access to agent information:
* none
* read-only
* read-write

More information on how to configure access with the `SimpleAccessControlModel` class is provided below under the description of that class.

The authorizer instance can be obtained by using the `getAuthorizer()` call, for both the receiver and the agent. For example:

```
receiver.getAuthorizer ().getCommunities ();
```

authorizer.addCommunity (community)

Adds a community string to the receiver's community authorization list. Does nothing if the community is already in the list, ensuring there is only one occurence of any given community string in the list.

authorizer.getCommunity (community)

Returns a community string if it is stored in the receiver's community authorization list, otherwise returns `null`.

authorizer.getCommunities ()

Returns the receiver's community authorization list.

authorizer.deleteCommunity (community)

Deletes a community string from the receiver's community authorization list. Does nothing if the community is not in the list.

authorizer.addUser (user)

Adds a user to the receiver's user authorization list. If a user of the same name is in the list, this call deletes the existing user, and replaces it with the supplied user, ensuring that only one user with a given name will exist in the list. The user object is in the same format as that used for the `session.createV3Session()` call.

```
var user = {
    name: "elsa"
    level: snmp.SecurityLevel.authPriv,
    authProtocol: snmp.AuthProtocols.sha,
    authKey: "imlettingitgo",
    privProtocol: snmp.PrivProtocols.des,
    privKey: "intotheunknown"
};

receiver.getAuthorizer ().addUser (elsa);
```

authorizer.getUser (userName)

Returns a user object if a user with the supplied name is stored in the receiver's user authorization list, otherwise returns `null`.

authorizer.getUsers ()

Returns the receiver's user authorization list.

authorizer.deleteUser (userName)

Deletes a user from the receiver's user authorization list. Does nothing if the user
with the supplied name is not in the list.

authorizer.getAccessControlModelType ()

Returns the `snmp.AccessControlModelType` of this authorizer, which is one of:
- `snmp.AccessControlModelType.None`
- `snmp.AccessControlModelType.Simple`

authorizer.getAccessControlModel ()

Returns the access control model object:
- for a type of `snmp.AccessControlModelType.None` - returns null (as the access control check returns positive every time)
- for a type of `snmp.AccessControlModelType.Simple` - returns a `SimpleAccessControlModel` object

Simple Access Control Model

The `SimpleAccessControlModel` class can be optionally selected as the access control model used by an `Agent`. The
`SimpleAccessControlModel` provides basic three-level access control for a given community or user.
The access levels are specified in the snmp.AccessLevel constant:
* `snmp.AccessLevel.None` - no access is granted to the community or user
* `snmp.AccessLevel.ReadOnly` - access is granted for the community or user for Get, GetNext and GetBulk requests but not Set requests
* `snmp.AccessLevel.ReadWrite` - access is granted for the community or user for Get, GetNext, GetBulk and Set requests

The `SimpleAccessControlModel` is not created via a direct API call, but is created internally by an `Agent`'s
`Authorizer` singleton.
So an agent's access control model can be accessed with:

```
var acm = agent.getAuthorizer ().getAccessControlModel ();
```

Note that any community or user that is used in any of the API calls in this section must first be created in the agent's `Authorizer`,
otherwise the agent will fail the initial community/user list check that the authorizer performs.

When using the Simple Access Control Model, the default access level for a newly created community or user in the `Authorizer` is read-only.

Example use:

```
var agent = snmp.createAgent({
    accessControlModelType: snmp.AccessControlModelType.Simple
```

```
}, function (error, data) {
    // null callback for example brevity
});
var authorizer = agent.getAuthorizer ();
authorizer.addCommunity ("public");
authorizer.addCommunity ("private");
authorizer.addUser ({
    name: "fred",
    level: snmp.SecurityLevel.noAuthNoPriv
});
var acm = authorizer.getAccessControlModel ();
// Since read-only is the default, explicitly setting read-only access is not required - just shown here as an example
acm.setCommunityAccess ("public", snmp.AccessLevel.ReadOnly);
acm.setCommunityAccess ("private", snmp.AccessLevel.ReadWrite);
acm.setUserAccess ("fred", snmp.AccessLevel.ReadWrite);
```

simpleAccessControlModel.setCommunityAccess (community, accessLevel)

Grant the given community the given access level.

simpleAccessControlModel.removeCommunityAccess (community)

Remove all access for the given community.

simpleAccessControlModel.getCommunityAccessLevel (community)

Return the access level for the given community.

simpleAccessControlModel.getCommunitiesAccess ()

Return a list of all community access control entries defined by this access control model.

simpleAccessControlModel.setUserAccess (userName, accessLevel)

Grant the given user the given access level.

simpleAccessControlModel.removeUserAccess (userName)

Remove all access for the given user.

simpleAccessControlModel.getUserAccessLevel (userName)

Return the access level for the given user.

simpleAccessControlModel.getUsersAccess ()

Return a list of all user access control entries defined by this access control model.

Mib Module

An `Agent` instance, when created, in turn creates an instance of the `Mib` class.
An agent always has one and only one `Mib` instance. The agent's `Mib` instance is
accessed through the `agent.getMib ()` call.

The MIB is a tree structure that holds management information. Information is "addressed"
in the tree by a series of integers, which form an Object ID (OID) from the root of the
tree down.

There are only two kinds of data structures that hold data in a MIB:

* **scalar** data - the scalar variable is stored at a node in the MIB tree, and
the value of the variable is a single child node of the scalar variable node, always with
address "0". For example, the sysDescr scalar variable is located at "1.3.6.1.2.1.1.1".
The value of the sysDescr variable is stored at "1.3.6.1.2.1.1.1.0"

```
1.3.6.1.2.1.1.1   <= sysDescr (scalar variable)
1.3.6.1.2.1.1.1.0 = OctetString: MyAwesomeHost  <= sysDescr.0 (scalar variable value)
```

* **table** data - the SNMP table stores data in columns and rows. Typically, if a table
is stored at a node in the MIB, it has an "entry" object addressed as "1" directly
below the table OID. Directly below the "entry" is a list of columns, which are typically
numbered from "1" upwards. Directly below each column are a series of rows. In the simplest
case a row is "indexed" by a single column in the table, but a row index can be a series of
columns, or columns that give multiple integers (e.g. an IPv4 address has four integers to
its index), or both. Here is an example of the hierarchy of an SNMP table for part of the
ifTable:

```
1.3.6.1.2.1.2.2   <= ifTable (table)
1.3.6.1.2.1.2.2.1   <= ifEntry (table entry)
1.3.6.1.2.1.2.2.1.1   <= ifIndex (column 1)
1.3.6.1.2.1.2.2.1.1.1 = Integer: 1   <= ifIndex row 1 value = 1
1.3.6.1.2.1.2.2.1.1.2 = Integer: 2   <= ifIndex row 2 value = 2
```

On creation, an `Agent` instance creates a singleton instance of the `Mib` module. You can
then register a "provider" to the agent's `Mib` instance that gives an interface to either a scalar
data instance, or a table.

```
var myScalarProvider = {
    name: "sysDescr",
    type: snmp.MibProviderType.Scalar,
    oid: "1.3.6.1.2.1.1.1",
```

```
        scalarType: snmp.ObjectType.OctetString,
        handler: function (mibRequest) {
           // e.g. can update the MIB data before responding to the request here
           mibRequest.done ();
        }
    };
    var mib = agent.getMib ();
    mib.registerProvider (myScalarProvider);
    mib.setScalarValue ("sysDescr", "MyAwesomeHost");
```

This code first gives the definition of a scalar "provider". A further explanation of these fields is given in the `mib.registerProvider()` section. Importantly, the `name` field is the unique identifier of the provider, and is used to select the specific provider in subsequent API calls.

The `registerProvider()` call adds the provider to the list of providers that the MIB holds. Note that this call does not add the "oid" node to the MIB tree. The first call of `setScalarValue()` will add the instance OID "1.3.6.1.2.1.1.1.0" to the MIB tree, along with its value.

At this point, the agent will serve up the value of this MIB node when the instance OID "1.3.6.1.2.1.1.1.0" is queried via SNMP.

A table provider has a similar definition:

```
    var myTableProvider = {
        name: "smallIfTable",
        type: snmp.MibProviderType.Table,
        oid: "1.3.6.1.2.1.2.2.1",
        tableColumns: [
            {
                number: 1,
                name: "ifIndex",
                type: snmp.ObjectType.Integer
            },
            {
                number: 2,
                name: "ifDescr",
                type: snmp.ObjectType.OctetString
            },
            {
                number: 3,
                name: "ifType",
                type: snmp.ObjectType.Integer,
                constraints: {
                    enumeration: {
                        "1": "goodif",
                        "2": "averageif",
```

```
                "3": "badif"
            }
        }
    }
],
tableIndex: [
    {
        columnName: "ifIndex"
    }
]
};
var mib = agent.getMib ();
mib.registerProvider (myTableProvider);
mib.addTableRow ("smallIfTable", [1, "eth0", 6]);
```

Here, the provider definition takes two additions fields: `tableColumns` for the column defintions, and `tableIndex` for the columns used for row indexes. In the example the `tableIndex` is the `ifIndex` column. The `mib.registerProvider()` section has further details on the fields that make up the provider definition.

The `oid` must be that of the "table entry" node, not its parent "table" node e.g. for `ifTable`, the `oid` in the provider is "1.3.6.1.2.1.2.2.1" (the OID for `ifEntry`).

Note that there is no `handler` callback function in this particular example, so any interaction is directly between SNMP requests and MIB values with no other intervention.

snmp.createMib ()

The `createMib()` function instantiates and returns an instance of the `Mib` class. The new Mib does not have any nodes (except for a single root node) and does not have any registered providers.

Note that this is only usable for an agent, not an AgentX subagent. Since an agent instanciates a `Mib` instance on creation, this call is not needed in many scenarios. Two scenarios where it might be useful are:

* where you want to pre-populate a `Mib` instance with providers and scalar/tabular data before creating the `Agent` instance itself.
* where you want to swap out an agent's existing `Mib` instance for an entirely new one.

mib.registerProvider (definition)

Registers a provider definition with the MIB. Doesn't add anything to the MIB tree.

A provider definition has these fields:

* `name` *(mandatory)* - the name of the provider, which serves as a unique key to reference the

provider for getting and setting values
* `type` *(mandatory)* - must be either `snmp.MibProviderType.Scalar` or `snmp.MibProviderType.Table` (mandatory)
* `oid` *(mandatory)* - the OID where the provider is registered in the MIB tree. Note that this is **not** the "instance node" (the ".0" node), but the one above it. In this case, the provider registers at "1.3.6.1.2.1.1.1", to provide the value at "1.3.6.1.2.1.1.1.0".
* `scalarType` *(mandatory for scalar types)* - only relevant to scalar provider type, this give the type of the variable, selected from `snmp.ObjectType`
* `tableColumns` *(mandatory for table types)* - gives any array of column definition objects for the table. Each column object must have a unique `number`, a `name` and a `type` from `snmp.ObjectType`. A column object with type `ObjectType.Integer` can optionally contain a `constraints` object, the format and meaning of which is identical to that defined on a single scalar provider (see `constraints` below for the details on this).
* `tableIndex` *(optional for table types)* - gives an array of index entry objects used for row indexes. Use a single-element array for a single-column index, and multiple values for a composite index. An index entry object has a `columnName` field, and if the entry is in another provider's table, then include a `foreign` field with the name of the foreign table's provider. If the `tableAugments` field is absent, `tableIndex` is mandatory.
* `tableAugments` *(optional for table types)* - gives the name of another registered provider that this table "augments". This means that the index information is taken from the given provider's table, and doesn't exist in the local table's column definitions. If the `tableIndex` field is absent, `tableAugments` is mandatory i.e. one of `tableIndex` and `tableAugments` needs to be present to define the table index.
* `handler` *(optional)* - an optional callback function, which is called before the request to the MIB is made. This could update the MIB value(s) handled by this provider. If not given, the values are simply returned from (or set in) the MIB without any other processing. The callback function takes a `MibRequest` instance, which has a `done()` function. This must be called when finished processing the request. The `MibRequest` also has an `oid` field with the instance OID being operated on, and an `operation` field with the request type from `snmp.PduType`. If the `MibRequest` is for a `SetRequest` PDU, then variables `setValue` and `setType` contain the value and type received in the `SetRequest` varbind.
* `constraints` *(optional for scalar types)* - an optional object to specify constraints for integer-based enumerated types. The only supported constraint at the moment is an `enumeration` object, which maps integers to their named types to capture "named-number enumerations" as described in RFC 2578 Section 7.1.1. Any SetRequest protocol operations are checked against the defined constraints, and are not actioned if the value in the SetRequest would violate the constraints e.g. the value is not a member of the defined enumeration. Note that table columns can specify such `constraints` in an identical way, except that these are stored under the column object definition for each column.

After registering the provider with the MIB, the provider is referenced by its `name` in other API calls.

While this call registers the provider to the MIB, it does not alter the MIB tree.

mib.registerProviders ([definitions])

Convenience method to register an array of providers in one call. Simply calls `registerProvider()` for each provider definition in the array.

mib.unregisterProvider (name)

Unregisters a provider from the MIB. This also deletes all MIB nodes from the provider's `oid` down the tree. It will also do upstream MIB tree pruning of any interior MIB nodes that only existed for the MIB tree to reach the provider `oid` node.

mib.getProviders ()

Returns an object of provider definitions registered with the MIB, indexed by provider name.

mib.getProvider (name)

Returns a single registered provider object for the given name.

mib.getScalarValue (scalarProviderName)

Retrieves the value from a scalar provider.

mib.setScalarValue (scalarProviderName, value)

Sets the value for a scalar provider. If this is the first time the scalar is set since the provider has registered with the MIB, it will also add the instance (".0") node and all required ancestors to the MIB tree.

mib.addTableRow (tableProviderName, row)

Adds a table row - in the form of an array of values - to a table provider. If the table is empty, this instantiates the provider's `oid` node and ancestors, its columns, before adding the row of values. Note that the row is an array of elements in the order of the table columns. If the table has any foreign index columns (i.e. those not belonging to this table), then values for these must be included the at the start of the row array, in the order they appear in the MIB INDEX clause.

mib.getTableColumnDefinitions (tableProviderName)

Returns a list of column definition objects for the provider.

mib.getTableCells (tableProviderName, byRow, includeInstances)

Returns a two-dimensional array of the table data. If `byRow` is false (the default), then the table data is given in a list of column arrays i.e. by column. If `byRow` is `true`, then the data is instead a list of row arrays. If `includeInstances` is `true`, then, for the column view there will be an extra first column with instance index information. If `includeInstances` is `true` for the row view, then there is an addition element at the start of each row with index information.

mib.getTableColumnCells (tableProviderName, columnNumber, includeInstances)

Returns a single column of table data for the given column number. If `includeInstances`
is `true`, then two arrays are returned: the first with instance index information,
and the second with the column data.

mib.getTableRowCells (tableProviderName, rowIndex)

Returns a single row of table data for the given row index. The row index is an array
of index values built from the node immediately under the column down to the node at the end
of the row instance, which will be a leaf node in the MIB tree. Ultimately, non-integer values
need to be converted to a sequence of integers that form the instance part of the OID. Here
are the details of the conversions from index values to row instance OID sequences:
- **ObjectType.Integer** - single integer
- **ObjectType.OctetString** - a sequence of integer ASCII values
- **ObjectType.OID** - the exact sequence of integers in the OID
- **ObjectType.IpAddress** - a sequence of the four integers in the IP address

mib.getTableSingleCell (tableProviderName, columnIndex, rowIndex)

Returns a single cell value from the column and row specified. The row index array is specified
in the same way as for the `getTableRowCells()` call.

mib.setTableSingleCell (tableProviderName, columnIndex, rowIndex, value)

Sets a single cell value at the column and row specified. The row index array is specified
in the same way as for the `getTableRowCells()` call.

mib.deleteTableRow (tableProviderName, rowIndex)

Deletes a table row at the row index specified. The row index array is specified
in the same way as for the `getTableRowCells()` call. If this was the last row in the table,
the table is pruned from the MIB, although the provider still remains registered with the MIB.
Meaning that on the addition of another row, the table will be instantiated again.

mib.dump (options)

Dumps the MIB in text format. The `options` object controls the display of the dump with these
options fields (all are booleans that default to `true`):

* `leavesOnly` - don't show interior nodes separately - only as prefix parts of leaf nodes
(instance nodes)
* `showProviders` - show nodes where providers are attached to the MIB
* `showTypes` - show instance value types
* `showValues` - show instance values

For example:

    mib.dump ();

produces this sort of output:

```
1.3.6.1.2.1.1.1 [Scalar: sysDescr]
1.3.6.1.2.1.1.1.0 = OctetString: Rage inside the machine!
1.3.6.1.2.1.2.2.1 [Table: ifTable]
1.3.6.1.2.1.2.2.1.1.1 = Integer: 1
1.3.6.1.2.1.2.2.1.1.2 = Integer: 2
1.3.6.1.2.1.2.2.1.2.1 = OctetString: lo
1.3.6.1.2.1.2.2.1.2.2 = OctetString: eth0
1.3.6.1.2.1.2.2.1.3.1 = Integer: 24
1.3.6.1.2.1.2.2.1.3.2 = Integer: 6
```

Using This Module: Module Store

The library supports MIB parsing by providing an interface to a `ModuleStore` instance into which you can load MIB modules from files, and fetch the resulting JSON MIB module representations.

Additionally, once a MIB is loaded into the module store, you can produce a list of MIB "provider" definitions that an `Agent` can register (see the `Agent` documentation for more details), so that you can start manipulating all the values defined in your MIB file right away.

```
// Create a module store, load a MIB module, and fetch its JSON representation
var store = snmp.createModuleStore ();
store.loadFromFile ("/path/to/your/mibs/SNMPv2-MIB.mib");
var jsonModule = store.getModule ("SNMPv2-MIB");

// Fetch MIB providers, create an agent, and register the providers with your agent
var providers = store.getProvidersForModule ("SNMPv2-MIB");
// Not recommended - but authorization and callback turned off for example brevity
var agent = snmp.createAgent ({disableAuthorization: true}, function (error, data) {});
var mib = agent.getMib ();
mib.registerProviders (providers);

// Start manipulating the MIB through the registered providers using the `Mib` API calls
mib.setScalarValue ("sysDescr", "The most powerful system you can think of");
mib.setScalarValue ("sysName", "multiplied-by-six");
mib.addTableRow ("sysOREntry", [1, "1.3.6.1.4.1.47491.42.43.44.45", "I've dreamed up this MIB", 20]);

// Then hit those bad boys with your favourite SNMP tools (or library ;-), e.g.
snmpwalk -v 2c -c public localhost 1.3.6.1
```

Meaning you can get right to the implementation of your MIB functionality with a minimum of boilerplate code.

snmp.createModuleStore ()

Creates a new `ModuleStore` instance, which comes pre-loaded with some "base" MIB modules that that provide MIB definitions that other MIB modules commonly refer to ("import"). The list of pre-loaded "base" modules is:

* RFC1155-SMI
* RFC1158-MIB
* RFC-1212
* RFC1213-MIB
* SNMPv2-SMI
* SNMPv2-CONF
* SNMPv2-TC
* SNMPv2-MIB

store.loadFromFile (fileName)

Loads all MIB modules in the given file into the module store. By convention, there is typically only a single MIB module per file, but there can be multiple module definitions stored in a single file. Loaded MIB modules are then referred to by this API by their MIB module name, not the source file name. The MIB module name is the name preceding the `DEFINITIONS ::= BEGIN` in the MIB file, and is often the very first thing present in a MIB file.

Note that if your MIB depends on ("imports") definitions from other MIB files, these must be loaded first e.g. the popular **IF-MIB** uses definitions from the **IANAifType-MIB**, which therefore must be loaded first. These dependencies are listed in the **IMPORTS** section of a MIB module, usually near the top of a MIB file. The pre-loaded "base" MIB modules contain many of the commonly used imports.

store.getModule (moduleName)

Retrieves the named MIB module from the store as a JSON object.

store.getModules (includeBase)

Retrieves all MIB modules from the store. If the `includeBase` boolean is set to true, then the base MIB modules are included in the list. The modules are returned as a single JSON "object of objects", keyed on the module name, with the values being entire JSON module represenations.

store.getModuleNames (includeBase)

Retrieves a list of the names of all MIB modules loaded in the store. If the `includeBase` boolean is set to true, then the base MIB modules names are included in the list.

store.getProvidersForModule (moduleName)

Returns an array of `Mib` "provider" definitions corresponding to all scalar and table instance objects contained in the named MIB module. The list of provider definitions are then ready to be registered to an agent's MIB by using the `agent.getMib().registerProviders()` call.

Forwarder Module

An `Agent` instance, when created, in turn creates an instance of the `Forwarder` class. There is no direct API call to create a `Forwarder` instance; this creation is the responsibility of the agent. An agent always has one and only one `Forwarder` instance. The agent's `Forwarder` instance is accessed through the `agent.getForwarder ()` call.

A `Forwader` is what RFC 3413 terms a "Proxy Forwarder Application". It maintains a list of "proxy" entries, each of which configures a named SNMPv3 context name to enable access to a given target host with the given user credentials. The `Forwarder` supports proxying of SNMPv3 sessions only.

```
var forwarder = agent.getForwarder ();
forwarder.addProxy({
  context: "slatescontext",
  host: "bedrock",
  user: {
     name: "slate",
     level: snmp.SecurityLevel.authNoPriv,
     authProtocol: snmp.AuthProtocols.sha,
     authKey: "quarryandgravel"
  },
});
```

Now requests to the agent with the context "slatescontext" supplied will be forwarded to host "bedrock", with the supplied credentials for user "slate".

You can query the proxy with a local agent user (added with the agent's `Authorizer` instance). Assuming your proxy runs on localhost, port 161, you could add local user "fred", and access the proxy with the new "fred" user.

```
var authorizer = agent.getAuthorizer();
authorizer.addUser ({
  name: "fred",
  level: snmp.SecurityLevel.noAuthNoPriv
});

// Test access using Net-SNMP tools (-n is the context option):

snmpget -v 3 -u fred -l noAuthNoPriv -n slatescontext localhost 1.3.6.1.2.1.1.1.0
```

```

This proxies requests through to "bedrock" as per the proxy definition.

## forwarder.addProxy (proxy)

Adds a new proxy to the forwarder.  The proxy is an object with these fields.

* `context` *(mandatory)* - the name of the SNMPv3 context for this proxy entry.  This is the unique key
  for proxy entries i.e. there cannot be two proxies with the same context name.
* `transport` *(optional)* - specifies the transport to use to reach the remote target.  Can be either
  `udp4` or `udp6`, defaults to `udp4`.
* `target` *(mandatory)* - the remote host that will receive proxied requests.
* `port` *(optional)* - the port of the SNMP agent on the remote host.  Defaults to 161.
* `user` *(mandatory)* - the SNMPv3 user.  The format for the user is described in the `createV3Session()`
  call documentation.

## forwarder.deleteProxy (context)

Delete the proxy for the given context from the forwarder.

## forwarder.getProxy (context)

Returns the forwarder's proxy for the given context.

## forwarder.getProxies ()

Returns an object containing a list of all registered proxies, keyed by context name.

## forwarder.dumpProxies ()

Prints a dump of all proxy definitions to the console.

# Using This Module: AgentX Subagent

The AgentX subagent implements the functionality specified in RFC 2741 to become a "subagent"
of an AgentX "master agent".  The goal of AgentX is to extend the functionality of an existing
"master" SNMP agent by a separate "subagent" registering parts of the MIB tree that it would
like to manage for the master agent.

The AgentX subagent supports the generation of all but two of the "administrative" PDU types, all of
which are sent from the subagent to the master agent:
* **Open PDU** - opens a new session with a master agent
* **Close PDU** - closes an existing session with the master agent
* **Register PDU** - registers a MIB region to control with the master agent
* **Unregister PDU** - unregisters a previously registered MIB region with the master agent
* **Notify PDU** - sends a notification to the master agent
* **Ping PDU** - sends a "ping" to confirm the master agent is still available
```

* **AddAgentCaps PDU** - adds an agent capability to the master agent's sysORTable
* **RemoveAgentCaps PDU** - remove a previously added agent capability from the master agent's sysORTable

The two unsupported "administrative" PDU types are:
* **IndexAllocate PDU** - request allocation of an index from a table whose index is managed by a master agent
* **IndexDeallocate PDU** - request deallocation of a previously allocated index from a master agent's table

These are unsupported as they do not fit the current MIB provider registration model, which only supports registering scalars and entire tables. These could be supported in the future by further generalizing the registration model to support table row registration.

The subagent responds to all "request processing" PDU types relevant to a Command Responder application, which are received from the master agent:

* **Get PDU** - requests exactly matched OID instances
* **GetNext PDU** - requests lexicographically "next" OID instances in the MIB tree
* **GetBulk PDU** - requests a series of "next" OID instances in the MIB tree
* **TestSet PDU** - tests a list of "set" operations to be committed as a single transaction
* **CommitSet PDU** - commits a list of "set" operations as a single transaction
* **UndoSet PDU** - undoes a list of "set" operations as a single transaction
* **CleanupSet PDU** - ends a "set" transaction

As per RFC 2741, all of these except the **CleanupSet** PDU return a **Response** PDU to the master agent.

Like the SNMP agent, the AgentX subagent maintains is a `Mib` instance, the API of which is detailed in the [Mib Module](#mib-module) section above. The subagent allows the MIB to be queried and manipulated through the API, as well as queried and manipulated through the AgentX interface with the above "request processing" PDUs (which are produced by the master agent when its SNMP interface is invoked).

It is important that MIB providers are registered using the subagent's `subagent.registerProvider ()` call (outlined below), and not using `subagent.getMib ().registerProvider ()`, as the subagent needs to both register the provider on its internal `Mib` object, *and* send a Register PDU to the master agent for the provider's MIB region. The latter step is skipped if registering the provider directly on the MIB object.

snmp.createSubagent (options)

The `createSubagent ()` function instantiates and returns an instance of the `Subagent` class:

```
// Default options
var options = {
    master: localhost
    masterPort: 705,
    timeout: 0,
    description: "Node net-snmp AgentX sub-agent",
};
```

subagent = snmp.createSubagent (options);

The `options` parameter is a mandatory object, possibly empty, and can contain the following fields:

* `master` - the host name or IP address of the master agent, which the subagent connects to.
* `masterPort` - the TCP port for the subagent to connect to the master agent on - defaults to 705.
* `timeout` - set the session-wide timeout on the master agent - defaults to 0, which means no session-wide timeout is set.
* `description` - a textual description of the subagent.

subagent.getMib ()

Returns the agent's singleton `Mib` instance, which is automatically created on creation of the subagent, and which holds all of the management data for the subagent.

subagent.open (callback)

Sends an `Open` PDU to the master agent to open a new session, invoking the callback on response from the master.

subagent.close (callback)

Sends a `Close` PDU to the master agent to close the subagent's session to the master, invoking the callback on response from the master.

subagent.registerProvider (provider, callback)

See the `Mib` class `registerProvider()` call for the definition of a provider. The format and meaning of the `provider` object is the same for this call. This sends a `Register` PDU to the master to register a region of the MIB for which the master will send "request processing" PDUs to the subagent. The supplied `callback` is used only once, on reception of the subsequent `Response` PDU from the master to the `Register` PDU. This is not to be confused with the `handler` optional callback on the provider definition, which is invoked for any "request processing" PDU received by the subagent for MIB objects in the registered MIB region.

subagent.unregisterProvider (name, callback)

Unregisters a previously registered MIB region by the supplied name of the provider. Sends an `Unregister` PDU to the master agent to do this. The supplied `callback` is used only once, on reception of the subsequent `Response` PDU from the master to the `Unregister` PDU.

subagent.registerProviders ([definitions], callback)

Convenience method to register an array of providers in one call. Simply calls `registerProvider()` for each provider definition in the array. The `callback` function is called once for each provider registered.

subagent.getProviders ()

Returns an object of provider definitions registered with the MIB, indexed by provider name.

subagent.getProvider (name)

Returns a single registered provider object for the given name.

subagent.addAgentCaps (oid, descr, callback)

Adds an agent capability - consisting of `oid` and `descr` - to the master agent's sysORTable.
Sends an `AddAgentCaps` PDU to the master to do this. The supplied `callback` is called on
reception of the subsequent `Response` PDU from the master to the `AddAgentCaps` PDU.

subagent.removeAgentCaps (oid, callback)

Remove an previously added capability from the master agent's sysORTable. Sends a `RemoveAgentCaps`
PDU to the master to do this. The supplied `callback` is called on reception of the subsequent
`Response` PDU from the master to the `RemoveAgentCaps` PDU.

subagent.notify (typeOrOid, varbinds, callback)

Sends a notification to the master agent using a `Notify` PDU. The notification takes the same
form as outlined in the `session.inform()` section above and also in RFC 2741 Section 6.2.10,
which is creating two varbinds that are always included in the notification:
- sysUptime.0 (1.3.6.1.2.1.1.3.0) - containing the subagent's uptime
- snmpTrapOID.0 (1.3.6.1.6.3.1.1.4.1.0) - containing the supplied OID (or supplied `snmp.TrapType` value)

The optional `varbinds` list is an additional list of varbind objects to append to the above
two varbinds. The supplied `callback` is called on reception of the subsequent
`Response` PDU from the master to the `Notify` PDU.

subagent.ping (callback)

Sends a "ping" to the master agent using a `Ping` PDU, to confirm that the master agent is still
responsive. The supplied `callback` is called on reception of the subsequent
`Response` PDU from the master to the `Ping` PDU.

Example Programs

Example programs are included under the module's `example` directory.

Changes

Version 1.0.0 - 14/01/2013

* Initial release including only SNMP version 1 support

Version 1.1.0 - 20/01/2013

* Implement SNMP version 2c support

Version 1.1.1 - 21/01/2013

* Correct name used in example `require()` call to include this module

Version 1.1.2 - 22/01/2013

* Implement `subtree()`, `table()` and `walk()` methods
* Support IPv6 (added `transport` option to the `createSession()` function)
* Re-order some methods in README.md

Version 1.1.3 - 27/01/2013

* Fix some typos and grammar errors in README.md
* Example `snmp-table` program had `snmp-subtree` in its usage message
* Implement example `snmp-tail` program to constantly poll for an OIDs value
* Add note to README.md about the ability to stop the `walk()` and `subtree()`
  methods by returning `true`

Version 1.1.4 - 29/01/2013

* Fix incorrect usage of the term "NPM" in README.md, should be "npm"

Version 1.1.5 - 05/02/2013

* The `transport` option to `createSession()` was not used

Version 1.1.6 - 12/04/2013

* Implement `tableColumns()` method
* Added example program `snmp-table-columns.js`
* Correct name of the `table` parameter to the `table()` callback
* Slight OID comparison performance enhancement

Version 1.1.7 - 11/05/2013

* Use MIT license instead of GPL

Version 1.1.8 - 22/06/2013

* Added the example program `cisco-device-inventory.js`
* Receive `Trap failed: TypeError: value is out of bounds` when sending

traps using SNMP version 2c

Version 1.1.9 - 03/11/2013

* Corrected a few instances of the parameter named `requestCallback` to some
  methods in the README.md file which should have been `feedCallback`
* Null type is used for varbinds with a 0 value
* Correct instances of snmp.Type to snmp.ObjectType in the README.md file

Version 1.1.10 - 01/12/2013

* Error handler in the `dgram.send()` callback in the `send()` method was
  creating a new instance of the `Error` class from the `error` parameter, but
  it was already an instance of the `Error` class (thanks Ray Solomon)
* Add stack traces to Error classes exported by this module (thanks Ray
  Solomon)
* Allow users to specify `0` retries when creating a session (thanks Ray
  Solomon)
* Update the list of SNMP version 1 related RFCs we adhere to in the
  `Standards Compliance` section of the README.md file

Version 1.1.11 - 27/12/2013

* Add `sourceAddress` and `sourcePort` optional options to the
  `Session` classes `createSession()` method, which can be used to control
  from which IP address and port messages should be sent
* Allow users to specify sysUpTime for SNMP traps and informs

Version 1.1.12 - 02/04/2014

* The `agentAddr` attribute is not used when passed in the `options` object
  to the `trap()` method

Version 1.1.13 - 12/08/2014

* Not catching error events for the UDP socket returned from the
  `dgram.createSocket()` function
* Some request methods do not copy arguments which results in sometimes
  unexpected behaviour
* Use a single UDP socket for all requests in a single SNMP session
* Use a try/catch block in the timer callback in the `Session.send()` method
* The `Session` can now emit an `error` event to catch errors in a sessions
  underlying UDP socket
* The `Session` can now emit a `close` event to catch close events from a
  sessions underlying UDP socket, which results in the cancellation of
  all outstanding requests
* Added a `close()` method to `Session` to close a sessions underlying UDP
  socket, which results a `close` event

* Signed integers are treated as unsigned integers when parsing response
  messages

Version 1.1.14 - 22/09/2015

* Host repository on GitHub

Version 1.1.15 - 08/02/2016

* When parsing an invalid response an exception in message parsing does not
  interupt response processing
* Incorrectly passing `req` object in call to `req.responseCb` when handling
  errors during response processing

Version 1.1.16 - 29/02/2016

* Address a number of issues detected with the Mocha test suite by a user

Version 1.1.17 - 21/03/2016

* Correct reference to non-existant `req` variable in the `Session` objects
  constructor (should be `this`)

Version 1.1.18 - 15/05/2015

* Correct argument number and names to the `snmp.createSession()` function
* Add missing braces to an example in the README.md file

Version 1.1.19 - 26/08/2016

* Remove 64bit integer check to ensure a maximum of 8 bytes are given in send
  and received messages

Version 1.2.0 - 22/07/2017

* Replace asn1 dependancy with asn1-ber

Version 1.2.1 - 11/02/2018

* Add support of 16bit ids to help interoperate with older devices (added the
  `idBitsSize` option to the `createSession()` function
* Add note to README.md that sessions should be closed when done with

Version 1.2.3 - 06/06/2018

* Set NoSpaceships Ltd to be the owner and maintainer

Version 1.2.4 - 07/06/2018

* Remove redundant sections from README.md

Version 2.0.0 - 16/01/2020

* Add SNMPv3 support

Version 2.1.0 - 16/01/2020

* Add trap and inform receiver

Version 2.1.1 - 17/01/2020

* Add CONTRIBUTING.md guidelines

Version 2.1.2 - 17/01/2020

* Add SNMPv3 context to Session class

Version 2.1.3 - 18/01/2020

* Add IPv6 option for tests

Version 2.2.0 - 21/01/2020

* Add SNMP agent

Version 2.3.0 - 22/01/2020

* Add MIB parser and module store

Version 2.4.0 - 24/01/2020

* Add proxy forwarder to agent

Version 2.5.0 - 25/01/2020

* Add AES-128 encryption

Version 2.5.1 - 27/01/2020

* Add non-integer, composite key, foreign key and augmented table index handling

Version 2.5.2 - 29/01/2020

* Update CONTRIBUTING.md and parser example

Version 2.5.3 - 22/02/2020

* Add backoff option

Version 2.5.4 - 22/03/2020

* Fix agent crash with unexpected GetNext start OID

Version 2.5.5 - 31/03/2020

* Fix double report PDU time synchronisation handling

Version 2.5.6 - 02/04/2020

* Fix agent handling of GetNext from off-tree OID

Version 2.5.7 - 09/04/2020

* Handle periodic report PDUs on a long running session

Version 2.5.8 - 13/04/2020

* Fix OID and namespace calculations in MIB parser

Version 2.5.9 - 17/04/2020

* Fix Windows absolute path for reading MIB files

Version 2.5.10 - 17/04/2020

* Improve SNMPv3 error messages

Version 2.5.11 - 21/04/2020

* Receiver close fix and receiver example fix

Version 2.5.12 - 24/04/2020

* Add backwardsGetNexts option for handling of errant GetNexts

Version 2.6.0 - 27/04/2020

* Add AgentX subagent

Version 2.6.1 - 02/05/2020

* Fix backwardsGetNexts session option and fix null MIB entry reading

Version 2.6.2 - 05/05/2020

* Add missing agent.close() API call

Version 2.6.3 - 07/05/2020

* Add set value to MibRequest and fix backwardsGetNexts

Version 2.6.4 - 09/05/2020

* Improve socket error handling

Version 2.6.5 - 26/05/2020

* Add agent support for handling short OIDs and noSuchInstance

Version 2.6.6 - 29/05/2020

* Fix async mibRequest handler

Version 2.6.7 - 01/06/2020

* Add support for zero-index rows in agent tables

Version 2.6.8 - 08/07/2020

* Fix GetBulk async mibRequest handling

Version 2.7.0 - 09/07/2020

* Add MIB create, add MIB setting for agent, and fix MIB error response crash

Version 2.7.1 - 17/07/2020

* Fix AgentX subagent noSuchInstance crash

Version 2.7.2 - 02/09/2020

* Declare variables to fix transpile errors

Version 2.7.3 - 02/09/2020

* MIB getobject callback convention update

Version 2.7.4 - 02/09/2020

* Fix columnNumber check in getColumnProvider

Version 2.7.5 - 05/09/2020

* Fix parsing of iso.org

Version 2.7.6 - 05/09/2020

* Add revisions/descriptions MIB parsing

Version 2.7.7 - 07/09/2020

* Fix double callback invocation on callback error

Version 2.8.0 - 09/09/2020

* Add eslint rules and conformance, fix AgentX subagent Unregister

Version 2.8.1 - 09/09/2020

* Add Travis CI configuration

Version 2.9.0 - 12/09/2020

* Add simple access control model for agent

Version 2.9.1 - 17/09/2020

* Add MIB integer enumeration constraints for providers and SetRequests

License

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/README.md
No license file was found, but licenses were detected in source scan.

Network Working Group Editor of this version:
Request for Comments: 3417 R. Presuhn
STD: 62 BMC Software, Inc.
Obsoletes: 1906 Authors of previous version:
Category: Standards Track J. Case
SNMP Research, Inc.
K. McCloghrie
Cisco Systems, Inc.
M. Rose
Dover Beach Consulting, Inc.
S. Waldbusser
International Network Services
December 2002

Transport Mappings for
the Simple Network Management Protocol (SNMP)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document defines the transport of Simple Network Management Protocol (SNMP) messages over various protocols. This document obsoletes RFC 1906.

Presuhn, et al. Standards Track [Page 1]

RFC 3417 Transport Mappings for SNMP December 2002

Table of Contents

1. Introduction

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB

module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

This document, Transport Mappings for the Simple Network Management Protocol, defines how the management protocol [RFC3416] may be carried over a variety of protocol suites. It is the purpose of this document to define how the SNMP maps onto an initial set of transport domains. At the time of this writing, work was in progress to define an IPv6 mapping, described in [RFC3419]. Other mappings may be defined in the future.

Although several mappings are defined, the mapping onto UDP over IPv4 is the preferred mapping for systems supporting IPv4. Systems implementing IPv4 MUST implement the mapping onto UDP over IPv4. To maximize interoperability, systems supporting other mappings SHOULD also provide for access via the UDP over IPv4 mapping.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

2. Definitions

```
SNMPv2-TM DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-IDENTITY,
    snmpModules, snmpDomains, snmpProxys
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION
        FROM SNMPv2-TC;

snmpv2tm MODULE-IDENTITY
    LAST-UPDATED "200210160000Z"
    ORGANIZATION "IETF SNMPv3 Working Group"
    CONTACT-INFO
        "WG-EMail:   snmpv3@lists.tislabs.com
         Subscribe:  snmpv3-request@lists.tislabs.com

         Co-Chair:   Russ Mundy
                     Network Associates Laboratories
         postal:     15204 Omega Drive, Suite 300
                     Rockville, MD 20850-4601
                     USA
         EMail:      mundy@tislabs.com
         phone:      +1 301 947-7107
```



```
         Co-Chair:   David Harrington
                     Enterasys Networks
         postal:     35 Industrial Way
                     P. O. Box 5005
                     Rochester, NH 03866-5005
                     USA
         EMail:      dbh@enterasys.com
         phone:      +1 603 337-2614

         Editor:     Randy Presuhn
                     BMC Software, Inc.
         postal:     2141 North First Street
                     San Jose, CA 95131
                     USA
```

```
        EMail:   randy_presuhn@bmc.com
        phone:   +1 408 546-1006"
    DESCRIPTION
        "The MIB module for SNMP transport mappings.

         Copyright (C) The Internet Society (2002). This
         version of this MIB module is part of RFC 3417;
         see the RFC itself for full legal notices.
        "
    REVISION     "200210160000Z"
    DESCRIPTION
        "Clarifications, published as RFC 3417."
    REVISION    "199601010000Z"
    DESCRIPTION
        "Clarifications, published as RFC 1906."
    REVISION    "199304010000Z"
    DESCRIPTION
        "The initial version, published as RFC 1449."
    ::= { snmpModules 19 }

-- SNMP over UDP over IPv4

snmpUDPDomain  OBJECT-IDENTITY
    STATUS     current
    DESCRIPTION
        "The SNMP over UDP over IPv4 transport domain.
        The corresponding transport address is of type
        SnmpUDPAddress."
    ::= { snmpDomains 1 }
```

Presuhn, et al.   Standards Track   [Page 4]

RFC 3417   Transport Mappings for SNMP   December 2002

```
SnmpUDPAddress ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "1d.1d.1d.1d/2d"
    STATUS       current
    DESCRIPTION
        "Represents a UDP over IPv4 address:
```

```
          octets   contents        encoding
           1-4     IP-address      network-byte order
           5-6     UDP-port        network-byte order
         "
    SYNTAX       OCTET STRING (SIZE (6))

-- SNMP over OSI

snmpCLNSDomain OBJECT-IDENTITY
    STATUS     current
    DESCRIPTION
         "The SNMP over CLNS transport domain.
         The corresponding transport address is of type
         SnmpOSIAddress."
    ::= { snmpDomains 2 }

snmpCONSDomain OBJECT-IDENTITY
    STATUS     current
    DESCRIPTION
         "The SNMP over CONS transport domain.
         The corresponding transport address is of type
         SnmpOSIAddress."
    ::= { snmpDomains 3 }

SnmpOSIAddress ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "*1x:/1x:"
    STATUS       current
    DESCRIPTION
         "Represents an OSI transport-address:

       octets   contents            encoding
          1     length of NSAP     'n' as an unsigned-integer
                                      (either 0 or from 3 to 20)
       2..(n+1) NSAP                concrete binary representation
       (n+2)..m TSEL                string of (up to 64) octets
         "
    SYNTAX       OCTET STRING (SIZE (1 | 4..85))
```

```
-- SNMP over DDP

snmpDDPDomain  OBJECT-IDENTITY
    STATUS     current
    DESCRIPTION
           "The SNMP over DDP transport domain.  The corresponding
           transport address is of type SnmpNBPAddress."
    ::= { snmpDomains 4 }

SnmpNBPAddress ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
           "Represents an NBP name:

      octets        contents          encoding
       1           length of object  'n' as an unsigned integer
      2..(n+1)      object            string of (up to 32) octets
       n+2         length of type    'p' as an unsigned integer
   (n+3)..(n+2+p)   type              string of (up to 32) octets
      n+3+p        length of zone    'q' as an unsigned integer
  (n+4+p)..(n+3+p+q) zone              string of (up to 32) octets

           For comparison purposes, strings are
           case-insensitive. All strings may contain any octet
           other than 255 (hex ff)."
    SYNTAX       OCTET STRING (SIZE (3..99))

-- SNMP over IPX

snmpIPXDomain  OBJECT-IDENTITY
    STATUS     current
    DESCRIPTION
           "The SNMP over IPX transport domain.  The corresponding
           transport address is of type SnmpIPXAddress."
    ::= { snmpDomains 5 }

SnmpIPXAddress ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "4x.1x:1x:1x:1x:1x:1x.2d"
    STATUS       current
    DESCRIPTION
           "Represents an IPX address:

             octets   contents            encoding
              1-4     network-number      network-byte order
              5-10    physical-address    network-byte order
             11-12    socket-number       network-byte order
```

"

SYNTAX OCTET STRING (SIZE (12))


```
-- for proxy to SNMPv1 (RFC 1157)

rfc1157Proxy   OBJECT IDENTIFIER ::= { snmpProxys 1 }

rfc1157Domain  OBJECT-IDENTITY
    STATUS     deprecated
    DESCRIPTION
            "The transport domain for SNMPv1 over UDP over IPv4.
            The corresponding transport address is of type
            SnmpUDPAddress."
    ::= { rfc1157Proxy 1 }

--  ::= { rfc1157Proxy 2 }            this OID is obsolete

END
```

3. SNMP over UDP over IPv4

This is the preferred transport mapping.

3.1. Serialization

Each instance of a message is serialized (i.e., encoded according to the convention of [BER]) onto a single UDP [RFC768] over IPv4 [RFC791] datagram, using the algorithm specified in Section 8.

3.2. Well-known Values

It is suggested that administrators configure their SNMP entities supporting command responder applications to listen on UDP port 161. Further, it is suggested that SNMP entities supporting notification receiver applications be configured to listen on UDP port 162.

When an SNMP entity uses this transport mapping, it must be capable of accepting messages up to and including 484 octets in size. It is recommended that implementations be capable of accepting messages of up to 1472 octets in size. Implementation of larger values is encouraged whenever possible.

4. SNMP over OSI

This is an optional transport mapping.

4.1. Serialization

Each instance of a message is serialized onto a single TSDU [IS8072] [IS8072A] for the OSI Connectionless-mode Transport Service (CLTS), using the algorithm specified in Section 8.

4.2. Well-known Values

It is suggested that administrators configure their SNMP entities supporting command responder applications to listen on transport selector "snmp-l" (which consists of six ASCII characters), when using a CL-mode network service to realize the CLTS. Further, it is suggested that SNMP entities supporting notification receiver applications be configured to listen on transport selector "snmpt-l" (which consists of seven ASCII characters, six letters and a hyphen) when using a CL-mode network service to realize the CLTS. Similarly, when using a CO-mode network service to realize the CLTS, the suggested transport selectors are "snmp-o" and "snmpt-o", for command responders and notification receivers, respectively.

When an SNMP entity uses this transport mapping, it must be capable of accepting messages that are at least 484 octets in size. Implementation of larger values is encouraged whenever possible.

5. SNMP over DDP

This is an optional transport mapping.

5.1. Serialization

Each instance of a message is serialized onto a single DDP datagram [APPLETALK], using the algorithm specified in Section 8.

5.2. Well-known Values

SNMP messages are sent using DDP protocol type 8. SNMP entities supporting command responder applications listen on DDP socket number 8, while SNMP entities supporting notification receiver applications listen on DDP socket number 9.

Administrators must configure their SNMP entities supporting command responder applications to use NBP type "SNMP Agent" (which consists of ten ASCII characters) while those supporting notification receiver applications must be configured to use NBP type "SNMP Trap Handler" (which consists of seventeen ASCII characters).

The NBP name for SNMP entities supporting command responders and notification receivers should be stable - NBP names should not change any more often than the IP address of a typical TCP/IP node. It is suggested that the NBP name be stored in some form of stable storage.

When an SNMP entity uses this transport mapping, it must be capable of accepting messages that are at least 484 octets in size. Implementation of larger values is encouraged whenever possible.

5.3. Discussion of AppleTalk Addressing

The AppleTalk protocol suite has certain features not manifest in the TCP/IP suite. AppleTalk's naming strategy and the dynamic nature of address assignment can cause problems for SNMP entities that wish to manage AppleTalk networks. TCP/IP nodes have an associated IP address which distinguishes each from the other. In contrast, AppleTalk nodes generally have no such characteristic. The network-level address, while often relatively stable, can change at every reboot (or more frequently).

Thus, when SNMP is mapped over DDP, nodes are identified by a "name", rather than by an "address". Hence, all AppleTalk nodes that implement this mapping are required to respond to NBP lookups and confirms (e.g., implement the NBP protocol stub), which guarantees that a mapping from NBP name to DDP address will be possible.

In determining the SNMP identity to register for an SNMP entity, it is suggested that the SNMP identity be a name which is associated with other network services offered by the machine.

NBP lookups, which are used to map NBP names into DDP addresses, can cause large amounts of network traffic as well as consume CPU resources. It is also the case that the ability to perform an NBP lookup is sensitive to certain network disruptions (such as zone table inconsistencies) which would not prevent direct AppleTalk communications between two SNMP entities.

Thus, it is recommended that NBP lookups be used infrequently, primarily to create a cache of name-to-address mappings. These cached mappings should then be used for any further SNMP traffic. It is recommended that SNMP entities supporting command generator applications should maintain this cache between reboots. This caching can help minimize network traffic, reduce CPU load on the network, and allow for (some amount of) network trouble shooting when the basic name-to-address translation mechanism is broken.

5.3.1. How to Acquire NBP names

An SNMP entity supporting command generator applications may have a pre-configured list of names of "known" SNMP entities supporting command responder applications. Similarly, an SNMP entity supporting command generator or notification receiver applications might interact with an operator. Finally, an SNMP entity supporting command generator or notification receiver applications might communicate with all SNMP entities supporting command responder or notification originator applications in a set of zones or networks.

5.3.2. When to Turn NBP names into DDP addresses

When an SNMP entity uses a cache entry to address an SNMP packet, it should attempt to confirm the validity mapping, if the mapping hasn't been confirmed within the last T1 seconds. This cache entry lifetime, T1, has a minimum, default value of 60 seconds, and should be configurable.

An SNMP entity supporting a command generator application may decide to prime its cache of names prior to actually communicating with another SNMP entity. In general, it is expected that such an entity may want to keep certain mappings "more current" than other mappings, e.g., those nodes which represent the network infrastructure (e.g., routers) may be deemed "more important".

Note that an SNMP entity supporting command generator applications should not prime its entire cache upon initialization - rather, it should attempt resolutions over an extended period of time (perhaps in some pre-determined or configured priority order). Each of these resolutions might, in fact, be a wildcard lookup in a given zone.

An SNMP entity supporting command responder applications must never prime its cache. When generating a response, such an entity does not need to confirm a cache entry. An SNMP entity supporting notification originator applications should do NBP lookups (or confirms) only when it needs to send an SNMP trap or inform.

5.3.3. How to Turn NBP names into DDP addresses

If the only piece of information available is the NBP name, then an NBP lookup should be performed to turn that name into a DDP address. However, if there is a piece of stale information, it can be used as a hint to perform an NBP confirm (which sends a unicast to the network address which is presumed to be the target of the name lookup) to see if the stale information is, in fact, still valid.

An NBP name to DDP address mapping can also be confirmed implicitly using only SNMP transactions. For example, an SNMP entity supporting command generator applications issuing a retrieval operation could also retrieve the relevant objects from the NBP group [RFC1742] for the SNMP entity supporting the command responder application. This information can then be correlated with the source DDP address of the response.

5.3.4. What if NBP is broken

Under some circumstances, there may be connectivity between two SNMP entities, but the NBP mapping machinery may be broken, e.g.,

o the NBP FwdReq (forward NBP lookup onto local attached network) mechanism might be broken at a router on the other entity's network; or,

o the NBP BrRq (NBP broadcast request) mechanism might be broken at a router on the entity's own network; or, o NBP might be broken on the other entity's node.

An SNMP entity supporting command generator applications which is dedicated to AppleTalk management might choose to alleviate some of these failures by directly implementing the router portion of NBP. For example, such an entity might already know all the zones on the AppleTalk internet and the networks on which each zone appears. Given an NBP lookup which fails, the entity could send an NBP FwdReq to the network in which the SNMP entity supporting the command responder or notification originator application was last located. If that failed, the station could then send an NBP LkUp (NBP lookup packet) as a directed (DDP) multicast to each network number on that network. Of the above (single) failures, this combined approach will solve the case where either the local router's BrRq-to-FwdReq mechanism is broken or the remote router's FwdReq-to-LkUp mechanism is broken.

6. SNMP over IPX

This is an optional transport mapping.

6.1. Serialization

Each instance of a message is serialized onto a single IPX datagram [NOVELL], using the algorithm specified in Section 8.

6.2. Well-known Values

SNMP messages are sent using IPX packet type 4 (i.e., Packet Exchange Protocol).

It is suggested that administrators configure their SNMP entities supporting command responder applications to listen on IPX socket 36879 (900f hexadecimal). Further, it is suggested that those supporting notification receiver applications be configured to listen on IPX socket 36880 (9010 hexadecimal).

When an SNMP entity uses this transport mapping, it must be capable of accepting messages that are at least 546 octets in size. Implementation of larger values is encouraged whenever possible.

7. Proxy to SNMPv1

Historically, in order to support proxy to SNMPv1, as defined in [RFC2576], it was deemed useful to define a transport domain, rfc1157Domain, which indicates the transport mapping for SNMP messages as defined in [RFC1157].

8. Serialization using the Basic Encoding Rules

When the Basic Encoding Rules [BER] are used for serialization:

(1) When encoding the length field, only the definite form is used; use of the indefinite form encoding is prohibited. Note that when using the definite-long form, it is permissible to use more than the minimum number of length octets necessary to encode the length field.

(2) When encoding the value field, the primitive form shall be used for all simple types, i.e., INTEGER, OCTET STRING, and OBJECT IDENTIFIER (either IMPLICIT or explicit). The constructed form of encoding shall be used only for structured types, i.e., a SEQUENCE or an IMPLICIT SEQUENCE.

(3) When encoding an object whose syntax is described using the BITS construct, the value is encoded as an OCTET STRING, in which all the named bits in (the definition of) the bitstring, commencing with the first bit and proceeding to the last bit, are placed in bits 8 (high order bit) to 1 (low order bit) of the first octet, followed by bits 8 to 1 of each subsequent octet in turn, followed by as many bits as are needed of the final subsequent octet, commencing with bit 8. Remaining bits, if any, of the final octet are set to zero on generation and ignored on receipt.

These restrictions apply to all aspects of ASN.1 encoding, including the message wrappers, protocol data units, and the data objects they contain.

8.1. Usage Example

As an example of applying the Basic Encoding Rules, suppose one wanted to encode an instance of the GetBulkRequest-PDU [RFC3416]:

```
  [5] IMPLICIT SEQUENCE {
          request-id      1414684022,
          non-repeaters   1,
          max-repetitions 2,
          variable-bindings {
              { name sysUpTime,
                value { unSpecified NULL } },
              { name ipNetToMediaPhysAddress,
                value { unSpecified NULL } },
              { name ipNetToMediaType,
                value { unSpecified NULL } }
          }
      }
```

Applying the BER, this may be encoded (in hexadecimal) as:

```
[5] IMPLICIT SEQUENCE          a5 82 00 39
    INTEGER                    02 04 54 52 5d 76
    INTEGER                    02 01 01
    INTEGER                    02 01 02
    SEQUENCE (OF)              30 2b
        SEQUENCE               30 0b
            OBJECT IDENTIFIER  06 07 2b 06 01 02 01 01 03
            NULL               05 00
        SEQUENCE               30 0d
            OBJECT IDENTIFIER  06 09 2b 06 01 02 01 04 16 01 02
            NULL               05 00
        SEQUENCE               30 0d
            OBJECT IDENTIFIER  06 09 2b 06 01 02 01 04 16 01 04
            NULL               05 00
```

Note that the initial SEQUENCE in this example was not encoded using the minimum number of length octets. (The first octet of the length, 82, indicates that the length of the content is encoded in the next two octets.)

9. Notice on Intellectual Property

The IETF 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 this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

10. Acknowledgments

This document is the product of the SNMPv3 Working Group. Some special thanks are in order to the following Working Group members:

Randy Bush
Jeffrey D. Case
Mike Daniele

Rob Frye
Lauren Heintz
Keith McCloghrie
Russ Mundy
David T. Perkins
Randy Presuhn
Aleksey Romanov
Juergen Schoenwaelder
Bert Wijnen

This version of the document, edited by Randy Presuhn, was initially based on the work of a design team whose members were:

Jeffrey D. Case
Keith McCloghrie
David T. Perkins
Randy Presuhn
Juergen Schoenwaelder

The previous versions of this document, edited by Keith McCloghrie, was the result of significant work by four major contributors:

Jeffrey D. Case
Keith McCloghrie
Marshall T. Rose
Steven Waldbusser

Additionally, the contributions of the SNMPv2 Working Group to the previous versions are also acknowledged. In particular, a special thanks is extended for the contributions of:

Alexander I. Alten
Dave Arneson
Uri Blumenthal
Doug Book
Kim Curran
Jim Galvin
Maria Greene
Iain Hanson
Dave Harrington
Nguyen Hien

Jeff Johnson
Michael Kornegay
Deirdre Kostick
David Levi
Daniel Mahoney
Bob Natale
Brian O'Keefe
Andrew Pearson
Dave Perkins
Randy Presuhn
Aleksey Romanov
Shawn Routhier
Jon Saperia
Juergen Schoenwaelder
Bob Stewart
Kaj Tesink
Glenn Waters
Bert Wijnen

11. IANA Considerations

The SNMPv2-TM MIB module requires the allocation of a single object identifier for its MODULE-IDENTITY. IANA has allocated this object identifier in the snmpModules subtree, defined in the SNMPv2-SMI MIB module.

12. Security Considerations

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change) the objects accessible through a command responder application.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model STD 62, RFC 3414 [RFC3414] and the View-based Access Control Model STD 62, RFC 3415 [RFC3415] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to a MIB is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change) them.

13. References

13.1. Normative References

[BER] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8825, December 1987.

[IS8072] Information processing systems - Open Systems Interconnection - Transport Service Definition, International Organization for Standardization. International Standard 8072, June 1986.

[IS8072A] Information processing systems - Open Systems Interconnection - Transport Service Definition - Addendum 1: Connectionless-mode Transmission, International Organization for Standardization. International Standard 8072/AD 1, December 1986.

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3414] Blumenthal, U. and B. Wijnen, "The User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3415] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

[RFC3416] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

13.2. Informative References

[APPLETALK] Sidhu, G., Andrews, R. and A. Oppenheimer, Inside AppleTalk (second edition). Addison-Wesley, 1990.

[NOVELL] Network System Technical Interface Overview. Novell, Inc., June 1989.

[RFC1157] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[RFC1742] Waldbusser, S. and K. Frisa, "AppleTalk Management Information Base II", RFC 1742, January 1995.

[RFC2576] Frye, R., Levi, D., Routhier, S. and B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework", RFC 2576, March 2000.

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3419] Daniele, M. and J. Schoenwaelder, "Textual Conventions for Transport Addresses", RFC 3419, November 2002.

14. Changes from RFC 1906

This document differs from RFC 1906 only in editorial improvements. The protocol is unchanged.

15. Editor's Address

Randy Presuhn
BMC Software, Inc.
2141 North First Street
San Jose, CA 95131
USA

Phone: +1 408 546-1006
EMail: randy_presuhn@bmc.com

16. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Presuhn, et al.   Standards Track   [Page 19]

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/ref/rfc/v3/rfc3417.txt
No license file was found, but licenses were detected in source scan.

together what you may or may not have done, or spend time reverse engineering your prose.

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/CONTRIBUTING.md
No license file was found, but licenses were detected in source scan.

```
RFC1213-MIB DEFINITIONS ::= BEGIN

IMPORTS
       mgmt, NetworkAddress, IpAddress, Counter, Gauge,
               TimeTicks
           FROM RFC1155-SMI
       OBJECT-TYPE
               FROM RFC-1212;

--  This MIB module uses the extended OBJECT-TYPE macro as
--  defined in [14];

--  MIB-II (same prefix as MIB-I)

mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }

-- textual conventions

DisplayString ::=
   OCTET STRING
-- This data type is used to model textual information taken
-- from the NVT ASCII character set.  By convention, objects
-- with this syntax are declared as having
```

```
--
--      SIZE (0..255)

PhysAddress ::=
   OCTET STRING
-- This data type is used to model media addresses.  For many
-- types of media, this will be in a binary representation.
-- For example, an ethernet address would be represented as
-- a string of 6 octets.

-- groups in MIB-II

system       OBJECT IDENTIFIER ::= { mib-2 1 }

interfaces   OBJECT IDENTIFIER ::= { mib-2 2 }

at           OBJECT IDENTIFIER ::= { mib-2 3 }

ip           OBJECT IDENTIFIER ::= { mib-2 4 }

icmp         OBJECT IDENTIFIER ::= { mib-2 5 }

tcp          OBJECT IDENTIFIER ::= { mib-2 6 }

udp          OBJECT IDENTIFIER ::= { mib-2 7 }

egp          OBJECT IDENTIFIER ::= { mib-2 8 }

-- historical (some say hysterical)
-- cmot      OBJECT IDENTIFIER ::= { mib-2 9 }

transmission OBJECT IDENTIFIER ::= { mib-2 10 }

snmp         OBJECT IDENTIFIER ::= { mib-2 11 }

-- the System group

-- Implementation of the System group is mandatory for all
-- systems.  If an agent is not configured to have a value
-- for any of these variables, a string of length 0 is
-- returned.

sysDescr OBJECT-TYPE
   SYNTAX  DisplayString (SIZE (0..255))
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
```

```
        "A textual description of the entity.  This value
        should include the full name and version
        identification of the system's hardware type,
        software operating-system, and networking
        software.  It is mandatory that this only contain
        printable ASCII characters."
    ::= { system 1 }

sysObjectID OBJECT-TYPE
    SYNTAX  OBJECT IDENTIFIER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The vendor's authoritative identification of the
        network management subsystem contained in the
        entity.  This value is allocated within the SMI
        enterprises subtree (1.3.6.1.4.1) and provides an
        easy and unambiguous means for determining `what
        kind of box' is being managed.  For example, if
        vendor `Flintstones, Inc.' was assigned the
        subtree 1.3.6.1.4.1.4242, it could assign the
        identifier 1.3.6.1.4.1.4242.1.1 to its `Fred
        Router'."
    ::= { system 2 }

sysUpTime OBJECT-TYPE
    SYNTAX  TimeTicks
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The time (in hundredths of a second) since the
        network management portion of the system was last
        re-initialized."
    ::= { system 3 }

sysContact OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The textual identification of the contact person
        for this managed node, together with information
        on how to contact this person."
    ::= { system 4 }

sysName OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-write
```

```
  STATUS  mandatory
  DESCRIPTION
          "An administratively-assigned name for this
          managed node.  By convention, this is the node's
          fully-qualified domain name."
  ::= { system 5 }

sysLocation OBJECT-TYPE
  SYNTAX  DisplayString (SIZE (0..255))
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
          "The physical location of this node (e.g.,
          `telephone closet, 3rd floor')."
  ::= { system 6 }

sysServices OBJECT-TYPE
  SYNTAX  INTEGER (0..127)
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "A value which indicates the set of services that
          this entity primarily offers.

          The value is a sum.  This sum initially takes the
          value zero, Then, for each layer, L, in the range
          1 through 7, that this node performs transactions
          for, 2 raised to (L - 1) is added to the sum.  For
          example, a node which performs primarily routing
          functions would have a value of 4 (2^(3-1)).  In
          contrast, a node which is a host offering
          application services would have a value of 72
          (2^(4-1) + 2^(7-1)).  Note that in the context of
          the Internet suite of protocols, values should be
          calculated accordingly:

               layer  functionality
                   1  physical (e.g., repeaters)
                   2  datalink/subnetwork (e.g., bridges)
                   3  internet (e.g., IP gateways)
                   4  end-to-end  (e.g., IP hosts)
                   7  applications (e.g., mail relays)

          For systems including OSI protocols, layers 5 and
          6 may also be counted."
  ::= { system 7 }

-- the Interfaces group
```

```
-- Implementation of the Interfaces group is mandatory for
-- all systems.

ifNumber OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of network interfaces (regardless of
        their current state) present on this system."
  ::= { interfaces 1 }

-- the Interfaces table

-- The Interfaces table contains information on the entity's
-- interfaces.  Each interface is thought of as being
-- attached to a `subnetwork'.  Note that this term should
-- not be confused with `subnet' which refers to an
-- addressing partitioning scheme used in the Internet suite
-- of protocols.

ifTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF IfEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
        "A list of interface entries.  The number of
        entries is given by the value of ifNumber."
  ::= { interfaces 2 }

ifEntry OBJECT-TYPE
  SYNTAX  IfEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
        "An interface entry containing objects at the
        subnetwork layer and below for a particular
        interface."
  INDEX   { ifIndex }
  ::= { ifTable 1 }

IfEntry ::=
  SEQUENCE {
    ifIndex
        INTEGER,
    ifDescr
        DisplayString,
```

```
      ifType
         INTEGER,
      ifMtu
         INTEGER,
      ifSpeed
         Gauge,
      ifPhysAddress
         PhysAddress,
      ifAdminStatus
         INTEGER,
      ifOperStatus
         INTEGER,
      ifLastChange
         TimeTicks,
      ifInOctets
         Counter,
      ifInUcastPkts
         Counter,
      ifInNUcastPkts
         Counter,
      ifInDiscards
         Counter,
      ifInErrors
         Counter,
      ifInUnknownProtos
         Counter,
      ifOutOctets
         Counter,
      ifOutUcastPkts
         Counter,
      ifOutNUcastPkts
         Counter,
      ifOutDiscards
         Counter,
      ifOutErrors
         Counter,
      ifOutQLen
         Gauge,
      ifSpecific
         OBJECT IDENTIFIER
   }

ifIndex OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "A unique value for each interface.  Its value
```

```
            ranges between 1 and the value of ifNumber.  The
            value for each interface must remain constant at
            least from one re-initialization of the entity's
            network management system to the next re-
            initialization."
    ::= { ifEntry 1 }

ifDescr OBJECT-TYPE
    SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "A textual string containing information about the
            interface.  This string should include the name of
            the manufacturer, the product name and the version
            of the hardware interface."
    ::= { ifEntry 2 }

ifType OBJECT-TYPE
    SYNTAX  INTEGER {
                other(1),          -- none of the following
                regular1822(2),
                hdh1822(3),
                ddn-x25(4),
                rfc877-x25(5),
                ethernet-csmacd(6),
                iso88023-csmacd(7),
                iso88024-tokenBus(8),
                iso88025-tokenRing(9),
                iso88026-man(10),
                starLan(11),
                proteon-10Mbit(12),
                proteon-80Mbit(13),
                hyperchannel(14),
                fddi(15),
                lapb(16),
                sdlc(17),
                ds1(18),           -- T-1
                e1(19),            -- european equiv. of T-1
                basicISDN(20),
                primaryISDN(21),   -- proprietary serial
                propPointToPointSerial(22),
                ppp(23),
                softwareLoopback(24),
                eon(25),            -- CLNP over IP [11]
                ethernet-3Mbit(26),
                nsip(27),           -- XNS over IP
                slip(28),          -- generic SLIP
```

```
            ultra(29),          -- ULTRA technologies
            ds3(30),            -- T-3
            sip(31),            -- SMDS
            frame-relay(32)
        }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The type of interface, distinguished according to
        the physical/link protocol(s) immediately `below'
        the network layer in the protocol stack."
    ::= { ifEntry 3 }

ifMtu OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The size of the largest datagram which can be
        sent/received on the interface, specified in
        octets.  For interfaces that are used for
        transmitting network datagrams, this is the size
        of the largest network datagram that can be sent
        on the interface."
    ::= { ifEntry 4 }

ifSpeed OBJECT-TYPE
    SYNTAX  Gauge
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "An estimate of the interface's current bandwidth
        in bits per second.  For interfaces which do not
        vary in bandwidth or for those where no accurate
        estimation can be made, this object should contain
        the nominal bandwidth."
    ::= { ifEntry 5 }

ifPhysAddress OBJECT-TYPE
    SYNTAX  PhysAddress
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The interface's address at the protocol layer
        immediately `below' the network layer in the
        protocol stack.  For interfaces which do not have

        such an address (e.g., a serial line), this object
```

```
          should contain an octet string of zero length."
   ::= { ifEntry 6 }

ifAdminStatus OBJECT-TYPE
   SYNTAX  INTEGER {
              up(1),       -- ready to pass packets
              down(2),
              testing(3)   -- in some test mode
          }
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
          "The desired state of the interface.  The
          testing(3) state indicates that no operational
          packets can be passed."
   ::= { ifEntry 7 }

ifOperStatus OBJECT-TYPE
   SYNTAX  INTEGER {
              up(1),       -- ready to pass packets
              down(2),
              testing(3)   -- in some test mode
          }
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
          "The current operational state of the interface.
          The testing(3) state indicates that no operational
          packets can be passed."
   ::= { ifEntry 8 }

ifLastChange OBJECT-TYPE
   SYNTAX  TimeTicks
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
          "The value of sysUpTime at the time the interface
          entered its current operational state.  If the
          current state was entered prior to the last re-
          initialization of the local network management
          subsystem, then this object contains a zero
          value."
   ::= { ifEntry 9 }

ifInOctets OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
```

```
  DESCRIPTION
          "The total number of octets received on the
          interface, including framing characters."
  ::= { ifEntry 10 }

ifInUcastPkts OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of subnetwork-unicast packets
          delivered to a higher-layer protocol."
  ::= { ifEntry 11 }

ifInNUcastPkts OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of non-unicast (i.e., subnetwork-
          broadcast or subnetwork-multicast) packets
          delivered to a higher-layer protocol."
  ::= { ifEntry 12 }

ifInDiscards OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of inbound packets which were chosen
          to be discarded even though no errors had been
          detected to prevent their being deliverable to a
          higher-layer protocol.  One possible reason for
          discarding such a packet could be to free up
          buffer space."
  ::= { ifEntry 13 }

ifInErrors OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of inbound packets that contained
          errors preventing them from being deliverable to a
          higher-layer protocol."
  ::= { ifEntry 14 }

ifInUnknownProtos OBJECT-TYPE
```

```
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of packets received via the interface
            which were discarded because of an unknown or
            unsupported protocol."
    ::= { ifEntry 15 }

ifOutOctets OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of octets transmitted out of the
            interface, including framing characters."
    ::= { ifEntry 16 }

ifOutUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of packets that higher-level
            protocols requested be transmitted to a
            subnetwork-unicast address, including those that
            were discarded or not sent."
    ::= { ifEntry 17 }

ifOutNUcastPkts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of packets that higher-level
            protocols requested be transmitted to a non-
            unicast (i.e., a subnetwork-broadcast or
            subnetwork-multicast) address, including those
            that were discarded or not sent."
    ::= { ifEntry 18 }

ifOutDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of outbound packets which were chosen
```

```
        to be discarded even though no errors had been
        detected to prevent their being transmitted.  One
        possible reason for discarding such a packet could
        be to free up buffer space."
  ::= { ifEntry 19 }

ifOutErrors OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of outbound packets that could not be
        transmitted because of errors."
  ::= { ifEntry 20 }

ifOutQLen OBJECT-TYPE
  SYNTAX  Gauge
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The length of the output packet queue (in
        packets)."
  ::= { ifEntry 21 }

ifSpecific OBJECT-TYPE
  SYNTAX  OBJECT IDENTIFIER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "A reference to MIB definitions specific to the
        particular media being used to realize the
        interface.  For example, if the interface is
        realized by an ethernet, then the value of this
        object refers to a document defining objects
        specific to ethernet.  If this information is not
        present, its value should be set to the OBJECT
        IDENTIFIER { 0 0 }, which is a syntatically valid
        object identifier, and any conformant
        implementation of ASN.1 and BER must be able to
        generate and recognize this value."
  ::= { ifEntry 22 }

-- the Address Translation group

-- Implementation of the Address Translation group is
-- mandatory for all systems.  Note however that this group
-- is deprecated by MIB-II. That is, it is being included
```

```
-- solely for compatibility with MIB-I nodes, and will most
-- likely be excluded from MIB-III nodes.  From MIB-II and
-- onwards, each network protocol group contains its own
-- address translation tables.

-- The Address Translation group contains one table which is
-- the union across all interfaces of the translation tables
-- for converting a NetworkAddress (e.g., an IP address) into
-- a subnetwork-specific address.  For lack of a better term,
-- this document refers to such a subnetwork-specific address
-- as a `physical' address.

-- Examples of such translation tables are: for broadcast
-- media where ARP is in use, the translation table is
-- equivalent to the ARP cache; or, on an X.25 network where
-- non-algorithmic translation to X.121 addresses is
-- required, the translation table contains the
-- NetworkAddress to X.121 address equivalences.

atTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF AtEntry
  ACCESS  not-accessible
  STATUS  deprecated
  DESCRIPTION
        "The Address Translation tables contain the
        NetworkAddress to `physical' address equivalences.
        Some interfaces do not use translation tables for
        determining address equivalences (e.g., DDN-X.25
        has an algorithmic method); if all interfaces are
        of this type, then the Address Translation table
        is empty, i.e., has zero entries."
  ::= { at 1 }

atEntry OBJECT-TYPE
  SYNTAX  AtEntry
  ACCESS  not-accessible
  STATUS  deprecated
  DESCRIPTION
        "Each entry contains one NetworkAddress to
        `physical' address equivalence."
  INDEX   { atIfIndex,
          atNetAddress }
  ::= { atTable 1 }

AtEntry ::=
  SEQUENCE {
      atIfIndex
          INTEGER,
```

```
    atPhysAddress
        PhysAddress,
    atNetAddress
        NetworkAddress
}

atIfIndex OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  deprecated
  DESCRIPTION
        "The interface on which this entry's equivalence
        is effective.  The interface identified by a
        particular value of this index is the same
        interface as identified by the same value of
        ifIndex."
  ::= { atEntry 1 }

atPhysAddress OBJECT-TYPE
  SYNTAX  PhysAddress
  ACCESS  read-write
  STATUS  deprecated
  DESCRIPTION
        "The media-dependent `physical' address.

        Setting this object to a null string (one of zero
        length) has the effect of invaliding the
        corresponding entry in the atTable object.  That
        is, it effectively dissasociates the interface
        identified with said entry from the mapping
        identified with said entry.  It is an
        implementation-specific matter as to whether the
        agent removes an invalidated entry from the table.
        Accordingly, management stations must be prepared
        to receive tabular information from agents that
        corresponds to entries not currently in use.
        Proper interpretation of such entries requires
        examination of the relevant atPhysAddress object."
  ::= { atEntry 2 }

atNetAddress OBJECT-TYPE
  SYNTAX  NetworkAddress
  ACCESS  read-write
  STATUS  deprecated
  DESCRIPTION
        "The NetworkAddress (e.g., the IP address)
        corresponding to the media-dependent `physical'
        address."
```

```
  ::= { atEntry 3 }

-- the IP group

-- Implementation of the IP group is mandatory for all
-- systems.

ipForwarding OBJECT-TYPE
  SYNTAX  INTEGER {
              forwarding(1),    -- acting as a gateway
              not-forwarding(2) -- NOT acting as a gateway
          }
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
          "The indication of whether this entity is acting
          as an IP gateway in respect to the forwarding of
          datagrams received by, but not addressed to, this
          entity.  IP gateways forward datagrams.  IP hosts
          do not (except those source-routed via the host).

          Note that for some managed nodes, this object may
          take on only a subset of the values possible.
          Accordingly, it is appropriate for an agent to
          return a `badValue' response if a management
          station attempts to change this object to an
          inappropriate value."
  ::= { ip 1 }

ipDefaultTTL OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
          "The default value inserted into the Time-To-Live
          field of the IP header of datagrams originated at
          this entity, whenever a TTL value is not supplied
          by the transport layer protocol."
  ::= { ip 2 }

ipInReceives OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The total number of input datagrams received from
          interfaces, including those received in error."
  ::= { ip 3 }
```

```
ipInHdrErrors OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of input datagrams discarded due to
       errors in their IP headers, including bad
       checksums, version number mismatch, other format
       errors, time-to-live exceeded, errors discovered
       in processing their IP options, etc."
  ::= { ip 4 }

ipInAddrErrors OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of input datagrams discarded because
       the IP address in their IP header's destination
       field was not a valid address to be received at
       this entity.  This count includes invalid
       addresses (e.g., 0.0.0.0) and addresses of
       unsupported Classes (e.g., Class E).  For entities
       which are not IP Gateways and therefore do not
       forward datagrams, this counter includes datagrams
       discarded because the destination address was not
       a local address."
  ::= { ip 5 }

ipForwDatagrams OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of input datagrams for which this
       entity was not their final IP destination, as a
       result of which an attempt was made to find a
       route to forward them to that final destination.
       In entities which do not act as IP Gateways, this
       counter will include only those packets which were
       Source-Routed via this entity, and the Source-
       Route option processing was successful."
  ::= { ip 6 }

ipInUnknownProtos OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
```

```
  STATUS  mandatory
  DESCRIPTION
         "The number of locally-addressed datagrams
         received successfully but discarded because of an
         unknown or unsupported protocol."
  ::= { ip 7 }

ipInDiscards OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
         "The number of input IP datagrams for which no
         problems were encountered to prevent their
         continued processing, but which were discarded
         (e.g., for lack of buffer space).  Note that this
         counter does not include any datagrams discarded
         while awaiting re-assembly."
  ::= { ip 8 }

ipInDelivers OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
         "The total number of input datagrams successfully
         delivered to IP user-protocols (including ICMP)."
  ::= { ip 9 }

ipOutRequests OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
         "The total number of IP datagrams which local IP
         user-protocols (including ICMP) supplied to IP in
         requests for transmission.  Note that this counter
         does not include any datagrams counted in
         ipForwDatagrams."
  ::= { ip 10 }

ipOutDiscards OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
         "The number of output IP datagrams for which no
```

```
            problem was encountered to prevent their
            transmission to their destination, but which were
            discarded (e.g., for lack of buffer space).  Note
            that this counter would include datagrams counted
            in ipForwDatagrams if any such packets met this
            (discretionary) discard criterion."
    ::= { ip 11 }

ipOutNoRoutes OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of IP datagrams discarded because no
            route could be found to transmit them to their
            destination.  Note that this counter includes any
            packets counted in ipForwDatagrams which meet this
            `no-route' criterion.  Note that this includes any
            datagarms which a host cannot route because all of
            its default gateways are down."
    ::= { ip 12 }

ipReasmTimeout OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The maximum number of seconds which received
            fragments are held while they are awaiting
            reassembly at this entity."
    ::= { ip 13 }

ipReasmReqds OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of IP fragments received which needed
            to be reassembled at this entity."
    ::= { ip 14 }

ipReasmOKs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of IP datagrams successfully re-
            assembled."
```

```
  ::= { ip 15 }

ipReasmFails OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of failures detected by the IP re-
        assembly algorithm (for whatever reason: timed
        out, errors, etc).  Note that this is not
        necessarily a count of discarded IP fragments
        since some algorithms (notably the algorithm in
        RFC 815) can lose track of the number of fragments
        by combining them as they are received."
  ::= { ip 16 }

ipFragOKs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of IP datagrams that have been
        successfully fragmented at this entity."
  ::= { ip 17 }

ipFragFails OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of IP datagrams that have been
        discarded because they needed to be fragmented at
        this entity but could not be, e.g., because their
        Don't Fragment flag was set."
  ::= { ip 18 }

ipFragCreates OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of IP datagram fragments that have
        been generated as a result of fragmentation at
        this entity."
  ::= { ip 19 }

-- the IP address table
```

```
-- The IP address table contains this entity's IP addressing
-- information.

ipAddrTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF IpAddrEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
       "The table of addressing information relevant to
       this entity's IP addresses."
  ::= { ip 20 }

ipAddrEntry OBJECT-TYPE
  SYNTAX  IpAddrEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
       "The addressing information for one of this
       entity's IP addresses."
  INDEX   { ipAdEntAddr }
  ::= { ipAddrTable 1 }

IpAddrEntry ::=
  SEQUENCE {
    ipAdEntAddr
       IpAddress,
    ipAdEntIfIndex
       INTEGER,
    ipAdEntNetMask
       IpAddress,
    ipAdEntBcastAddr
       INTEGER,
    ipAdEntReasmMaxSize
       INTEGER (0..65535)
  }

ipAdEntAddr OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The IP address to which this entry's addressing
       information pertains."
  ::= { ipAddrEntry 1 }

ipAdEntIfIndex OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
```

```
  STATUS  mandatory
  DESCRIPTION
        "The index value which uniquely identifies the
        interface to which this entry is applicable.  The
        interface identified by a particular value of this
        index is the same interface as identified by the
        same value of ifIndex."
  ::= { ipAddrEntry 2 }

ipAdEntNetMask OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The subnet mask associated with the IP address of
        this entry.  The value of the mask is an IP
        address with all the network bits set to 1 and all
        the hosts bits set to 0."
  ::= { ipAddrEntry 3 }

ipAdEntBcastAddr OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The value of the least-significant bit in the IP
        broadcast address used for sending datagrams on
        the (logical) interface associated with the IP
        address of this entry.  For example, when the
        Internet standard all-ones broadcast address is
        used, the value will be 1.  This value applies to
        both the subnet and network broadcasts addresses
        used by the entity on this (logical) interface."
  ::= { ipAddrEntry 4 }

ipAdEntReasmMaxSize OBJECT-TYPE
  SYNTAX  INTEGER (0..65535)
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The size of the largest IP datagram which this
        entity can re-assemble from incoming IP fragmented
        datagrams received on this interface."
  ::= { ipAddrEntry 5 }

-- the IP routing table

-- The IP routing table contains an entry for each route
```

```
-- presently known to this entity.

ipRouteTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF IpRouteEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
       "This entity's IP Routing table."
  ::= { ip 21 }

ipRouteEntry OBJECT-TYPE
  SYNTAX  IpRouteEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
       "A route to a particular destination."
  INDEX   { ipRouteDest }
  ::= { ipRouteTable 1 }

IpRouteEntry ::=
  SEQUENCE {
    ipRouteDest
       IpAddress,
    ipRouteIfIndex
       INTEGER,
    ipRouteMetric1
       INTEGER,
    ipRouteMetric2
       INTEGER,
    ipRouteMetric3
       INTEGER,
    ipRouteMetric4
       INTEGER,
    ipRouteNextHop
       IpAddress,
    ipRouteType
       INTEGER,
    ipRouteProto
       INTEGER,
    ipRouteAge
       INTEGER,
    ipRouteMask
       IpAddress,
    ipRouteMetric5
       INTEGER,
    ipRouteInfo
       OBJECT IDENTIFIER
  }
```

```
ipRouteDest OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
       "The destination IP address of this route.  An
       entry with a value of 0.0.0.0 is considered a
       default route.  Multiple routes to a single
       destination can appear in the table, but access to
       such multiple entries is dependent on the table-
       access mechanisms defined by the network
       management protocol in use."
  ::= { ipRouteEntry 1 }

ipRouteIfIndex OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
       "The index value which uniquely identifies the
       local interface through which the next hop of this
       route should be reached.  The interface identified
       by a particular value of this index is the same
       interface as identified by the same value of
       ifIndex."
  ::= { ipRouteEntry 2 }

ipRouteMetric1 OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
       "The primary routing metric for this route.  The
       semantics of this metric are determined by the
       routing-protocol specified in the route's
       ipRouteProto value.  If this metric is not used,
       its value should be set to -1."
  ::= { ipRouteEntry 3 }

ipRouteMetric2 OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
       "An alternate routing metric for this route.  The
       semantics of this metric are determined by the
       routing-protocol specified in the route's
```

```
        ipRouteProto value.  If this metric is not used,
        its value should be set to -1."
  ::= { ipRouteEntry 4 }

ipRouteMetric3 OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
        "An alternate routing metric for this route.  The
        semantics of this metric are determined by the
        routing-protocol specified in the route's
        ipRouteProto value.  If this metric is not used,
        its value should be set to -1."
  ::= { ipRouteEntry 5 }

ipRouteMetric4 OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
        "An alternate routing metric for this route.  The
        semantics of this metric are determined by the
        routing-protocol specified in the route's
        ipRouteProto value.  If this metric is not used,
        its value should be set to -1."
  ::= { ipRouteEntry 6 }

ipRouteNextHop OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
        "The IP address of the next hop of this route.
        (In the case of a route bound to an interface
        which is realized via a broadcast media, the value
        of this field is the agent's IP address on that
        interface.)"
  ::= { ipRouteEntry 7 }

ipRouteType OBJECT-TYPE
  SYNTAX  INTEGER {
          other(1),        -- none of the following

          invalid(2),      -- an invalidated route

                           -- route to directly
          direct(3),       -- connected (sub-)network
```

```
                  -- route to a non-local
            indirect(4)      -- host/network/sub-network
          }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
          "The type of route.  Note that the values
          direct(3) and indirect(4) refer to the notion of
          direct and indirect routing in the IP
          architecture.

          Setting this object to the value invalid(2) has
          the effect of invalidating the corresponding entry
          in the ipRouteTable object.  That is, it
          effectively dissasociates the destination
          identified with said entry from the route
          identified with said entry.  It is an
          implementation-specific matter as to whether the
          agent removes an invalidated entry from the table.
          Accordingly, management stations must be prepared
          to receive tabular information from agents that
          corresponds to entries not currently in use.
          Proper interpretation of such entries requires
          examination of the relevant ipRouteType object."
    ::= { ipRouteEntry 8 }

  ipRouteProto OBJECT-TYPE
    SYNTAX  INTEGER {
            other(1),       -- none of the following

                  -- non-protocol information,
                  -- e.g., manually configured
            local(2),       -- entries

                  -- set via a network
            netmgmt(3),     -- management protocol

                  -- obtained via ICMP,
            icmp(4),        -- e.g., Redirect

                  -- the remaining values are
                  -- all gateway routing
                  -- protocols
            egp(5),
            ggp(6),
            hello(7),
            rip(8),
```

```
            is-is(9),
            es-is(10),
            ciscoIgrp(11),
            bbnSpfIgp(12),
            ospf(13),
            bgp(14)
          }
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
          "The routing mechanism via which this route was
          learned.  Inclusion of values for gateway routing
          protocols is not intended to imply that hosts
          should support those protocols."
    ::= { ipRouteEntry 9 }

ipRouteAge OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
          "The number of seconds since this route was last
          updated or otherwise determined to be correct.
          Note that no semantics of `too old' can be implied
          except through knowledge of the routing protocol
          by which the route was learned."
    ::= { ipRouteEntry 10 }

ipRouteMask OBJECT-TYPE
    SYNTAX  IpAddress
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
          "Indicate the mask to be logical-ANDed with the
          destination address before being compared to the
          value in the ipRouteDest field.  For those systems
          that do not support arbitrary subnet masks, an
          agent constructs the value of the ipRouteMask by
          determining whether the value of the correspondent
          ipRouteDest field belong to a class-A, B, or C
          network, and then using one of:

               mask           network
               255.0.0.0      class-A
               255.255.0.0    class-B
               255.255.255.0  class-C

          If the value of the ipRouteDest is 0.0.0.0 (a
```

```
          default route), then the mask value is also
          0.0.0.0.  It should be noted that all IP routing
          subsystems implicitly use this mechanism."
  ::= { ipRouteEntry 11 }

ipRouteMetric5 OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
          "An alternate routing metric for this route.  The
          semantics of this metric are determined by the
          routing-protocol specified in the route's
          ipRouteProto value.  If this metric is not used,
          its value should be set to -1."
  ::= { ipRouteEntry 12 }

ipRouteInfo OBJECT-TYPE
  SYNTAX  OBJECT IDENTIFIER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "A reference to MIB definitions specific to the
          particular routing protocol which is responsible
          for this route, as determined by the value
          specified in the route's ipRouteProto value.  If
          this information is not present, its value should
          be set to the OBJECT IDENTIFIER { 0 0 }, which is
          a syntatically valid object identifier, and any
          conformant implementation of ASN.1 and BER must be
          able to generate and recognize this value."
  ::= { ipRouteEntry 13 }

-- the IP Address Translation table

-- The IP address translation table contain the IpAddress to
-- `physical' address equivalences.  Some interfaces do not
-- use translation tables for determining address
-- equivalences (e.g., DDN-X.25 has an algorithmic method);
-- if all interfaces are of this type, then the Address
-- Translation table is empty, i.e., has zero entries.

ipNetToMediaTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF IpNetToMediaEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
          "The IP Address Translation table used for mapping
```

```
        from IP addresses to physical addresses."
  ::= { ip 22 }

ipNetToMediaEntry OBJECT-TYPE
  SYNTAX  IpNetToMediaEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
        "Each entry contains one IpAddress to `physical'
        address equivalence."
  INDEX   { ipNetToMediaIfIndex,
          ipNetToMediaNetAddress }
  ::= { ipNetToMediaTable 1 }

IpNetToMediaEntry ::=
  SEQUENCE {
     ipNetToMediaIfIndex
        INTEGER,
     ipNetToMediaPhysAddress
        PhysAddress,
     ipNetToMediaNetAddress
        IpAddress,
     ipNetToMediaType
        INTEGER
  }

ipNetToMediaIfIndex OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
        "The interface on which this entry's equivalence
        is effective.  The interface identified by a
        particular value of this index is the same
        interface as identified by the same value of
        ifIndex."
  ::= { ipNetToMediaEntry 1 }

ipNetToMediaPhysAddress OBJECT-TYPE
  SYNTAX  PhysAddress
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
        "The media-dependent `physical' address."
  ::= { ipNetToMediaEntry 2 }

ipNetToMediaNetAddress OBJECT-TYPE
  SYNTAX  IpAddress
```

```
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "The IpAddress corresponding to the media-
            dependent `physical' address."
    ::= { ipNetToMediaEntry 3 }

ipNetToMediaType OBJECT-TYPE
    SYNTAX  INTEGER {
                other(1),        -- none of the following
                invalid(2),      -- an invalidated mapping
                dynamic(3),
                static(4)
            }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
            "The type of mapping.

            Setting this object to the value invalid(2) has
            the effect of invalidating the corresponding entry
            in the ipNetToMediaTable.  That is, it effectively
            dissasociates the interface identified with said
            entry from the mapping identified with said entry.
            It is an implementation-specific matter as to
            whether the agent removes an invalidated entry
            from the table.  Accordingly, management stations
            must be prepared to receive tabular information
            from agents that corresponds to entries not
            currently in use.  Proper interpretation of such
            entries requires examination of the relevant
            ipNetToMediaType object."
    ::= { ipNetToMediaEntry 4 }

-- additional IP objects

ipRoutingDiscards OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of routing entries which were chosen
            to be discarded even though they are valid.  One
            possible reason for discarding such an entry could
            be to free-up buffer space for other routing

            entries."
    ::= { ip 23 }
```

```
-- the ICMP group

-- Implementation of the ICMP group is mandatory for all
-- systems.

icmpInMsgs OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
          "The total number of ICMP messages which the
          entity received.  Note that this counter includes
          all those counted by icmpInErrors."
   ::= { icmp 1 }

icmpInErrors OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
          "The number of ICMP messages which the entity
          received but determined as having ICMP-specific
          errors (bad ICMP checksums, bad length, etc.)."
   ::= { icmp 2 }

icmpInDestUnreachs OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
          "The number of ICMP Destination Unreachable
          messages received."
   ::= { icmp 3 }

icmpInTimeExcds OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
          "The number of ICMP Time Exceeded messages
          received."
   ::= { icmp 4 }

icmpInParmProbs OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
```

```
  DESCRIPTION
          "The number of ICMP Parameter Problem messages
          received."
  ::= { icmp 5 }

icmpInSrcQuenchs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Source Quench messages
          received."
  ::= { icmp 6 }

icmpInRedirects OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Redirect messages received."
  ::= { icmp 7 }

icmpInEchos OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Echo (request) messages
          received."
  ::= { icmp 8 }

icmpInEchoReps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Echo Reply messages received."
  ::= { icmp 9 }

icmpInTimestamps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Timestamp (request) messages
          received."
  ::= { icmp 10 }
```

```
icmpInTimestampReps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP Timestamp Reply messages
        received."
  ::= { icmp 11 }

icmpInAddrMasks OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP Address Mask Request messages
        received."
  ::= { icmp 12 }

icmpInAddrMaskReps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP Address Mask Reply messages
        received."
  ::= { icmp 13 }

icmpOutMsgs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of ICMP messages which this
        entity attempted to send.  Note that this counter
        includes all those counted by icmpOutErrors."
  ::= { icmp 14 }

icmpOutErrors OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP messages which this entity did
        not send due to problems discovered within ICMP

        such as a lack of buffers.  This value should not
        include errors discovered outside the ICMP layer
        such as the inability of IP to route the resultant
```

```
          datagram.  In some implementations there may be no
          types of error which contribute to this counter's
          value."
  ::= { icmp 15 }

icmpOutDestUnreachs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Destination Unreachable
          messages sent."
  ::= { icmp 16 }

icmpOutTimeExcds OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Time Exceeded messages sent."
  ::= { icmp 17 }

icmpOutParmProbs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Parameter Problem messages
          sent."
  ::= { icmp 18 }

icmpOutSrcQuenchs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Source Quench messages sent."
  ::= { icmp 19 }

icmpOutRedirects OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Redirect messages sent.  For a

          host, this object will always be zero, since hosts
          do not send redirects."
```

```
  ::= { icmp 20 }

icmpOutEchos OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP Echo (request) messages sent."
  ::= { icmp 21 }

icmpOutEchoReps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP Echo Reply messages sent."
  ::= { icmp 22 }

icmpOutTimestamps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP Timestamp (request) messages
        sent."
  ::= { icmp 23 }

icmpOutTimestampReps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP Timestamp Reply messages
        sent."
  ::= { icmp 24 }

icmpOutAddrMasks OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of ICMP Address Mask Request messages
        sent."
  ::= { icmp 25 }

icmpOutAddrMaskReps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
```

```
  STATUS  mandatory
  DESCRIPTION
          "The number of ICMP Address Mask Reply messages
          sent."
  ::= { icmp 26 }

-- the TCP group

-- Implementation of the TCP group is mandatory for all
-- systems that implement the TCP.

-- Note that instances of object types that represent
-- information about a particular TCP connection are
-- transient; they persist only as long as the connection
-- in question.

tcpRtoAlgorithm OBJECT-TYPE
  SYNTAX  INTEGER {
              other(1),    -- none of the following

              constant(2), -- a constant rto
              rsre(3),     -- MIL-STD-1778, Appendix B
              vanj(4)      -- Van Jacobson's algorithm [10]
          }
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The algorithm used to determine the timeout value
          used for retransmitting unacknowledged octets."
  ::= { tcp 1 }

tcpRtoMin OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The minimum value permitted by a TCP
          implementation for the retransmission timeout,
          measured in milliseconds.  More refined semantics
          for objects of this type depend upon the algorithm
          used to determine the retransmission timeout.  In
          particular, when the timeout algorithm is rsre(3),
          an object of this type has the semantics of the
          LBOUND quantity described in RFC 793."
  ::= { tcp 2 }

tcpRtoMax OBJECT-TYPE
  SYNTAX  INTEGER
```

```
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The maximum value permitted by a TCP
       implementation for the retransmission timeout,
       measured in milliseconds.  More refined semantics
       for objects of this type depend upon the algorithm
       used to determine the retransmission timeout.  In
       particular, when the timeout algorithm is rsre(3),
       an object of this type has the semantics of the
       UBOUND quantity described in RFC 793."
  ::= { tcp 3 }

tcpMaxConn OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The limit on the total number of TCP connections
       the entity can support.  In entities where the
       maximum number of connections is dynamic, this
       object should contain the value -1."
  ::= { tcp 4 }

tcpActiveOpens OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of times TCP connections have made a
       direct transition to the SYN-SENT state from the
       CLOSED state."
  ::= { tcp 5 }

tcpPassiveOpens OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of times TCP connections have made a
       direct transition to the SYN-RCVD state from the
       LISTEN state."
  ::= { tcp 6 }

tcpAttemptFails OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
```

```
    DESCRIPTION
            "The number of times TCP connections have made a
            direct transition to the CLOSED state from either
            the SYN-SENT state or the SYN-RCVD state, plus the
            number of times TCP connections have made a direct
            transition to the LISTEN state from the SYN-RCVD
            state."
    ::= { tcp 7 }

tcpEstabResets OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of times TCP connections have made a
            direct transition to the CLOSED state from either
            the ESTABLISHED state or the CLOSE-WAIT state."
    ::= { tcp 8 }

tcpCurrEstab OBJECT-TYPE
    SYNTAX  Gauge
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of TCP connections for which the
            current state is either ESTABLISHED or CLOSE-
            WAIT."
    ::= { tcp 9 }

tcpInSegs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of segments received, including
            those received in error.  This count includes
            segments received on currently established
            connections."
    ::= { tcp 10 }

tcpOutSegs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of segments sent, including
            those on current connections but excluding those
            containing only retransmitted octets."
```

```
  ::= { tcp 11 }

tcpRetransSegs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of segments retransmitted - that
       is, the number of TCP segments transmitted
       containing one or more previously transmitted
       octets."
  ::= { tcp 12 }

-- the TCP Connection table

-- The TCP connection table contains information about this
-- entity's existing TCP connections.

tcpConnTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF TcpConnEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
       "A table containing TCP connection-specific
       information."
  ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
  SYNTAX  TcpConnEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
       "Information about a particular current TCP
       connection.  An object of this type is transient,
       in that it ceases to exist when (or soon after)
       the connection makes the transition to the CLOSED
       state."
  INDEX   { tcpConnLocalAddress,
         tcpConnLocalPort,
         tcpConnRemAddress,
         tcpConnRemPort }
  ::= { tcpConnTable 1 }

TcpConnEntry ::=
  SEQUENCE {
    tcpConnState
       INTEGER,
    tcpConnLocalAddress
```

```
        IpAddress,
      tcpConnLocalPort
        INTEGER (0..65535),
      tcpConnRemAddress
        IpAddress,
      tcpConnRemPort
        INTEGER (0..65535)
    }

  tcpConnState OBJECT-TYPE
    SYNTAX  INTEGER {
            closed(1),
            listen(2),
            synSent(3),
            synReceived(4),
            established(5),
            finWait1(6),
            finWait2(7),
            closeWait(8),
            lastAck(9),
            closing(10),
            timeWait(11),
            deleteTCB(12)
          }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The state of this TCP connection.

        The only value which may be set by a management
        station is deleteTCB(12).  Accordingly, it is
        appropriate for an agent to return a `badValue'
        response if a management station attempts to set
        this object to any other value.

        If a management station sets this object to the
        value deleteTCB(12), then this has the effect of
        deleting the TCB (as defined in RFC 793) of the
        corresponding connection on the managed node,
        resulting in immediate termination of the
        connection.

        As an implementation-specific option, a RST

        segment may be sent from the managed node to the
        other TCP endpoint (note however that RST segments
        are not sent reliably)."
    ::= { tcpConnEntry 1 }
```

```
tcpConnLocalAddress OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The local IP address for this TCP connection.  In
       the case of a connection in the listen state which
       is willing to accept connections for any IP
       interface associated with the node, the value
       0.0.0.0 is used."
  ::= { tcpConnEntry 2 }

tcpConnLocalPort OBJECT-TYPE
  SYNTAX  INTEGER (0..65535)
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The local port number for this TCP connection."
  ::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The remote IP address for this TCP connection."
  ::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE
  SYNTAX  INTEGER (0..65535)
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The remote port number for this TCP connection."
  ::= { tcpConnEntry 5 }

-- additional TCP objects

tcpInErrs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of segments received in error
       (e.g., bad TCP checksums)."
  ::= { tcp 14 }
```

```
tcpOutRsts OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of TCP segments sent containing the
        RST flag."
  ::= { tcp 15 }

-- the UDP group

-- Implementation of the UDP group is mandatory for all
-- systems which implement the UDP.

udpInDatagrams OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of UDP datagrams delivered to
        UDP users."
  ::= { udp 1 }

udpNoPorts OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of received UDP datagrams for
        which there was no application at the destination
        port."
  ::= { udp 2 }

udpInErrors OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of received UDP datagrams that could
        not be delivered for reasons other than the lack
        of an application at the destination port."
  ::= { udp 3 }

udpOutDatagrams OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
```

```
        "The total number of UDP datagrams sent from this
        entity."
  ::= { udp 4 }

-- the UDP Listener table

-- The UDP listener table contains information about this
-- entity's UDP end-points on which a local application is
-- currently accepting datagrams.

udpTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF UdpEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
        "A table containing UDP listener information."
  ::= { udp 5 }

udpEntry OBJECT-TYPE
  SYNTAX  UdpEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
        "Information about a particular current UDP
        listener."
  INDEX   { udpLocalAddress, udpLocalPort }
  ::= { udpTable 1 }

UdpEntry ::=
  SEQUENCE {
    udpLocalAddress
        IpAddress,
    udpLocalPort
        INTEGER (0..65535)
  }

udpLocalAddress OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The local IP address for this UDP listener.  In

        the case of a UDP listener which is willing to
        accept datagrams for any IP interface associated
        with the node, the value 0.0.0.0 is used."
  ::= { udpEntry 1 }
```

```
udpLocalPort OBJECT-TYPE
  SYNTAX  INTEGER (0..65535)
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The local port number for this UDP listener."
  ::= { udpEntry 2 }

-- the EGP group

-- Implementation of the EGP group is mandatory for all
-- systems which implement the EGP.

egpInMsgs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of EGP messages received without
        error."
  ::= { egp 1 }

egpInErrors OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of EGP messages received that proved
        to be in error."
  ::= { egp 2 }

egpOutMsgs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of locally generated EGP
        messages."
  ::= { egp 3 }

egpOutErrors OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The number of locally generated EGP messages not
        sent due to resource limitations within an EGP
        entity."
```

```
  ::= { egp 4 }

-- the EGP Neighbor table

-- The EGP neighbor table contains information about this
-- entity's EGP neighbors.

egpNeighTable OBJECT-TYPE
  SYNTAX  SEQUENCE OF EgpNeighEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
        "The EGP neighbor table."
  ::= { egp 5 }

egpNeighEntry OBJECT-TYPE
  SYNTAX  EgpNeighEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
        "Information about this entity's relationship with
        a particular EGP neighbor."
  INDEX   { egpNeighAddr }
  ::= { egpNeighTable 1 }

EgpNeighEntry ::=
  SEQUENCE {
    egpNeighState
      INTEGER,
    egpNeighAddr
      IpAddress,
    egpNeighAs
      INTEGER,
    egpNeighInMsgs
      Counter,
    egpNeighInErrs
      Counter,
    egpNeighOutMsgs
      Counter,
    egpNeighOutErrs
      Counter,
    egpNeighInErrMsgs
      Counter,
    egpNeighOutErrMsgs
      Counter,
    egpNeighStateUps
      Counter,
    egpNeighStateDowns
```

```
        Counter,
      egpNeighIntervalHello
        INTEGER,
      egpNeighIntervalPoll
        INTEGER,
      egpNeighMode
        INTEGER,
      egpNeighEventTrigger
        INTEGER
  }

egpNeighState OBJECT-TYPE
  SYNTAX  INTEGER {
            idle(1),
            acquisition(2),
            down(3),
            up(4),
            cease(5)
        }
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The EGP state of the local system with respect to
        this entry's EGP neighbor.  Each EGP state is
        represented by a value that is one greater than
        the numerical value associated with said state in
        RFC 904."
  ::= { egpNeighEntry 1 }

egpNeighAddr OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The IP address of this entry's EGP neighbor."
  ::= { egpNeighEntry 2 }

egpNeighAs OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The autonomous system of this EGP peer.  Zero
        should be specified if the autonomous system
        number of the neighbor is not yet known."
  ::= { egpNeighEntry 3 }

egpNeighInMsgs OBJECT-TYPE
```

```
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of EGP messages received without error
       from this EGP peer."
  ::= { egpNeighEntry 4 }

egpNeighInErrs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of EGP messages received from this EGP
       peer that proved to be in error (e.g., bad EGP
       checksum)."
  ::= { egpNeighEntry 5 }

egpNeighOutMsgs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of locally generated EGP messages to
       this EGP peer."
  ::= { egpNeighEntry 6 }

egpNeighOutErrs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of locally generated EGP messages not
       sent to this EGP peer due to resource limitations
       within an EGP entity."
  ::= { egpNeighEntry 7 }

egpNeighInErrMsgs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The number of EGP-defined error messages received
       from this EGP peer."
  ::= { egpNeighEntry 8 }

egpNeighOutErrMsgs OBJECT-TYPE
  SYNTAX  Counter
```

```
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of EGP-defined error messages sent to
            this EGP peer."
    ::= { egpNeighEntry 9 }

egpNeighStateUps OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of EGP state transitions to the UP
            state with this EGP peer."
    ::= { egpNeighEntry 10 }

egpNeighStateDowns OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The number of EGP state transitions from the UP
            state to any other state with this EGP peer."
    ::= { egpNeighEntry 11 }

egpNeighIntervalHello OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The interval between EGP Hello command
            retransmissions (in hundredths of a second).  This
            represents the t1 timer as defined in RFC 904."
    ::= { egpNeighEntry 12 }

egpNeighIntervalPoll OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The interval between EGP poll command

            retransmissions (in hundredths of a second).  This
            represents the t3 timer as defined in RFC 904."
    ::= { egpNeighEntry 13 }

egpNeighMode OBJECT-TYPE
    SYNTAX  INTEGER { active(1), passive(2) }
```

```
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The polling mode of this EGP entity, either
          passive or active."
  ::= { egpNeighEntry 14 }

egpNeighEventTrigger OBJECT-TYPE
  SYNTAX  INTEGER { start(1), stop(2) }
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
          "A control variable used to trigger operator-
          initiated Start and Stop events.  When read, this
          variable always returns the most recent value that
          egpNeighEventTrigger was set to.  If it has not
          been set since the last initialization of the
          network management subsystem on the node, it
          returns a value of `stop'.

          When set, this variable causes a Start or Stop
          event on the specified neighbor, as specified on
          pages 8-10 of RFC 904.  Briefly, a Start event
          causes an Idle peer to begin neighbor acquisition
          and a non-Idle peer to reinitiate neighbor
          acquisition.  A stop event causes a non-Idle peer
          to return to the Idle state until a Start event
          occurs, either via egpNeighEventTrigger or
          otherwise."
  ::= { egpNeighEntry 15 }

-- additional EGP objects

egpAs OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
          "The autonomous system number of this EGP entity."
  ::= { egp 6 }

-- the Transmission group

-- Based on the transmission media underlying each interface
-- on a system, the corresponding portion of the Transmission
-- group is mandatory for that system.

-- When Internet-standard definitions for managing
```

```
-- transmission media are defined, the transmission group is
-- used to provide a prefix for the names of those objects.

-- Typically, such definitions reside in the experimental
-- portion of the MIB until they are "proven", then as a
-- part of the Internet standardization process, the
-- definitions are accordingly elevated and a new object
-- identifier, under the transmission group is defined. By
-- convention, the name assigned is:
--
--     type OBJECT IDENTIFIER    ::= { transmission number }
--
-- where "type" is the symbolic value used for the media in
-- the ifType column of the ifTable object, and "number" is
-- the actual integer value corresponding to the symbol.

-- the SNMP group

-- Implementation of the SNMP group is mandatory for all
-- systems which support an SNMP protocol entity.  Some of
-- the objects defined below will be zero-valued in those
-- SNMP implementations that are optimized to support only
-- those functions specific to either a management agent or
-- a management station.  In particular, it should be
-- observed that the objects below refer to an SNMP entity,
-- and there may be several SNMP entities residing on a
-- managed node (e.g., if the node is hosting acting as
-- a management station).

snmpInPkts OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of Messages delivered to the
       SNMP entity from the transport service."
  ::= { snmp 1 }

snmpOutPkts OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of SNMP Messages which were
       passed from the SNMP protocol entity to the
       transport service."
  ::= { snmp 2 }
```

```
snmpInBadVersions OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP Messages which were
        delivered to the SNMP protocol entity and were for
        an unsupported SNMP version."
  ::= { snmp 3 }

snmpInBadCommunityNames OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP Messages delivered to
        the SNMP protocol entity which used a SNMP
        community name not known to said entity."
  ::= { snmp 4 }

snmpInBadCommunityUses OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP Messages delivered to
        the SNMP protocol entity which represented an SNMP
        operation which was not allowed by the SNMP
        community named in the Message."
  ::= { snmp 5 }

snmpInASNParseErrs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of ASN.1 or BER errors
        encountered by the SNMP protocol entity when
        decoding received SNMP Messages."
  ::= { snmp 6 }

-- { snmp 7 } is not used

snmpInTooBigs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
```

```
        "The total number of SNMP PDUs which were
        delivered to the SNMP protocol entity and for
        which the value of the error-status field is
        `tooBig'."
  ::= { snmp 8 }

snmpInNoSuchNames OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP PDUs which were
        delivered to the SNMP protocol entity and for
        which the value of the error-status field is
        `noSuchName'."
  ::= { snmp 9 }

snmpInBadValues OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP PDUs which were
        delivered to the SNMP protocol entity and for
        which the value of the error-status field is
        `badValue'."
  ::= { snmp 10 }

snmpInReadOnlys OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number valid SNMP PDUs which were
        delivered to the SNMP protocol entity and for
        which the value of the error-status field is
        `readOnly'.  It should be noted that it is a
        protocol error to generate an SNMP PDU which
        contains the value `readOnly' in the error-status
        field, as such this object is provided as a means
        of detecting incorrect implementations of the

        SNMP."
  ::= { snmp 11 }

snmpInGenErrs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
```

```
    STATUS  mandatory
    DESCRIPTION
            "The total number of SNMP PDUs which were
            delivered to the SNMP protocol entity and for
            which the value of the error-status field is
            `genErr'."
    ::= { snmp 12 }

snmpInTotalReqVars OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of MIB objects which have been
            retrieved successfully by the SNMP protocol entity
            as the result of receiving valid SNMP Get-Request
            and Get-Next PDUs."
    ::= { snmp 13 }

snmpInTotalSetVars OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of MIB objects which have been
            altered successfully by the SNMP protocol entity
            as the result of receiving valid SNMP Set-Request
            PDUs."
    ::= { snmp 14 }

snmpInGetRequests OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of SNMP Get-Request PDUs which
            have been accepted and processed by the SNMP
            protocol entity."
    ::= { snmp 15 }

snmpInGetNexts OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "The total number of SNMP Get-Next PDUs which have
            been accepted and processed by the SNMP protocol
            entity."
```

```
  ::= { snmp 16 }

snmpInSetRequests OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of SNMP Set-Request PDUs which
       have been accepted and processed by the SNMP
       protocol entity."
  ::= { snmp 17 }

snmpInGetResponses OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of SNMP Get-Response PDUs which
       have been accepted and processed by the SNMP
       protocol entity."
  ::= { snmp 18 }

snmpInTraps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of SNMP Trap PDUs which have
       been accepted and processed by the SNMP protocol
       entity."
  ::= { snmp 19 }

snmpOutTooBigs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of SNMP PDUs which were
       generated by the SNMP protocol entity and for
       which the value of the error-status field is
       `tooBig.'"
  ::= { snmp 20 }

snmpOutNoSuchNames OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
```

```
        "The total number of SNMP PDUs which were
        generated by the SNMP protocol entity and for
        which the value of the error-status is
        `noSuchName'."
  ::= { snmp 21 }

snmpOutBadValues OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP PDUs which were
        generated by the SNMP protocol entity and for
        which the value of the error-status field is
        `badValue'."
  ::= { snmp 22 }

-- { snmp 23 } is not used

snmpOutGenErrs OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP PDUs which were
        generated by the SNMP protocol entity and for
        which the value of the error-status field is
        `genErr'."
  ::= { snmp 24 }

snmpOutGetRequests OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP Get-Request PDUs which
        have been generated by the SNMP protocol entity."
  ::= { snmp 25 }

snmpOutGetNexts OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The total number of SNMP Get-Next PDUs which have
        been generated by the SNMP protocol entity."
  ::= { snmp 26 }
```

```
snmpOutSetRequests OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of SNMP Set-Request PDUs which
       have been generated by the SNMP protocol entity."
  ::= { snmp 27 }

snmpOutGetResponses OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of SNMP Get-Response PDUs which
       have been generated by the SNMP protocol entity."
  ::= { snmp 28 }

snmpOutTraps OBJECT-TYPE
  SYNTAX  Counter
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
       "The total number of SNMP Trap PDUs which have
       been generated by the SNMP protocol entity."
  ::= { snmp 29 }

snmpEnableAuthenTraps OBJECT-TYPE
  SYNTAX  INTEGER { enabled(1), disabled(2) }
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
       "Indicates whether the SNMP agent process is
       permitted to generate authentication-failure
       traps.  The value of this object overrides any
       configuration information; as such, it provides a
       means whereby all authentication-failure traps may
       be disabled.

       Note that it is strongly recommended that this
       object be stored in non-volatile memory so that it
       remains constant between re-initializations of the
       network management system."
  ::= { snmp 30 }

END
```

Found in path(s):

* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/lib/mibs/RFC1213-MIB.mib
No license file was found, but licenses were detected in source scan.

Network Working Group J. Case
Request for Comments: 3412 SNMP Research, Inc.
STD: 62 D. Harrington
Obsoletes: 2572 Enterasys Networks
Category: Standards Track R. Presuhn
BMC Software, Inc.
B. Wijnen
Lucent Technologies
December 2002

Message Processing and Dispatching for the
Simple Network Management Protocol (SNMP)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document describes the Message Processing and Dispatching for Simple Network Management Protocol (SNMP) messages within the SNMP architecture. It defines the procedures for dispatching potentially multiple versions of SNMP messages to the proper SNMP Message Processing Models, and for dispatching PDUs to SNMP applications. This document also describes one Message Processing Model - the SNMPv3 Message Processing Model. This document obsoletes RFC 2572.

Table of Contents

1. Introduction

The Architecture for describing Internet Management Frameworks [RFC3411] describes that an SNMP engine is composed of:

1) a Dispatcher
2) a Message Processing Subsystem,
3) a Security Subsystem, and
4) an Access Control Subsystem.

Applications make use of the services of these subsystems.

It is important to understand the SNMP architecture and its terminology to understand where the Message Processing Subsystem and Dispatcher described in this document fit into the architecture and interact with other subsystems within the architecture. The reader is expected to have read and understood the description of the SNMP architecture, defined in [RFC3411].

The Dispatcher in the SNMP engine sends and receives SNMP messages. It also dispatches SNMP PDUs to SNMP applications. When an SNMP message needs to be prepared or when data needs to be extracted from an SNMP message, the Dispatcher delegates these tasks to a message version-specific Message Processing Model within the Message Processing Subsystem.

A Message Processing Model is responsible for processing an SNMP version-specific message and for coordinating the interaction with the Security Subsystem to ensure proper security is applied to the

SNMP message being handled.

Interactions between the Dispatcher, the Message Processing Subsystem, and applications are modeled using abstract data elements and abstract service interface primitives defined by the SNMP architecture.

Similarly, interactions between the Message Processing Subsystem and the Security Subsystem are modeled using abstract data elements and abstract service interface primitives as defined by the SNMP architecture.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119.

2. Overview

The following illustration depicts the Message Processing in relation to SNMP applications, the Security Subsystem and Transport Mappings.

```
+-------------------------------------------------------------------+
| SNMP Entity                                                       |
|                                                                   |
| +-------------------------------------------------------------+   |
| | Applications                                                |   |
| | +-----------+  +--------------+                             |   |
| | | Command   |  | Notification |                             |   |
| | | Generator |  | Originator   | +-----------+ +--------------+|   |
| | +-----------+  +--------------+ | Proxy     | | Other        ||   |
| | +-----------+  +--------------+ | Forwarder | |Application(s)||   |
| | | Command   |  | Notification | +-----------+ +--------------+|   |
| | | Responder |  | Receiver     |                             |   |
| | +-----------+  +--------------+                             |   |
| +-------------------------------------------------------------+   |
|        ^                ^               ^           ^             |
|        |                |               |           |             |
```

```
|       v              v             v          v           |
|       +--------+-------+---------------+-----------+            |
|               ^                                      |
|               |    +---------------------+  +-----------------+ |
|               |    | Message Processing  |  | Security        | |
| Dispatcher      v    | Subsystem           |  | Subsystem       | |
| +------------------+ |     +------------+  |  |                 | |
| | PDU Dispatcher   | |  +->| v1MP     * |<--->| +-------------+ | |
| |                  | |  |  +------------+  |  | | Other       | | |
| |                  | |  |  +------------+  |  | | Security    | | |
| |                  | |  +->| v2cMP    * |<--->| | Model       | | |
| | Message          | |  |  +------------+  |  | +-------------+ | |
| | Dispatcher  <-------->+                  |  |                 | |
| |                  | |  |  +------------+  |  | +-------------+ | |
| |                  | |  +->| v3MP     * |<--->| | User-based  | | |
| | Transport        | |  |  +------------+  |  | | Security    | | |
| | Mapping          | |  |  +------------+  |  | | Model       | | |
| | (e.g., RFC 3417) | |  +->| otherMP  * |<--->| +-------------+ | |
| +------------------+ |     +------------+  |  |                 | |
|         ^          +---------------------+  +-----------------+ |
|         |                                              |
+----------|-----------------------------------------------------+
          v
  +------------------+
  |   Network        |          * One or more models may be present.
  +------------------+
```


2.1. The Dispatcher

The Dispatcher is a key piece of an SNMP engine. There is only one in an SNMP engine, and its job is to dispatch tasks to the multiple version-specific Message Processing Models, and to dispatch PDUs to various applications.

For outgoing messages, an application provides a PDU to be sent, plus the data needed to prepare and send the message, and the application specifies which version-specific Message Processing Model will be used to prepare the message with the desired security processing. Once the message is prepared, the Dispatcher sends the message.

For incoming messages, the Dispatcher determines the SNMP version of the incoming message and passes the message to the version-specific Message Processing Model to extract the components of the message and to coordinate the processing of security services for the message. After version-specific processing, the PDU Dispatcher determines which application, if any, should receive the PDU for processing and forwards it accordingly.

The Dispatcher, while sending and receiving SNMP messages, collects statistics about SNMP messages and the behavior of the SNMP engine in managed objects to make them accessible to remote SNMP entities. This document defines these managed objects, the MIB module which contains them, and how these managed objects might be used to provide useful management.

2.2. Message Processing Subsystem

The SNMP Message Processing Subsystem is the part of an SNMP engine which interacts with the Dispatcher to handle the version-specific SNMP messages. It contains one or more Message Processing Models.

This document describes one Message Processing Model, the SNMPv3 Message Processing Model, in Section 6. The SNMPv3 Message Processing Model is defined in a separate section to show that multiple (independent) Message Processing Models can exist at the same time and that such Models can be described in different documents. The SNMPv3 Message Processing Model can be replaced or supplemented with other Message Processing Models in the future. Two Message Processing Models which are expected to be developed in the future are the SNMPv1 message format [RFC1157] and the SNMPv2c message format [RFC1901]. Others may be developed as needed.

3. Elements of Message Processing and Dispatching

See [RFC3411] for the definitions of:

contextEngineID

contextName
scopedPDU
maxSizeResponseScopedPDU
securityModel
securityName
securityLevel
messageProcessingModel

For incoming messages, a version-specific message processing module provides these values to the Dispatcher. For outgoing messages, an application provides these values to the Dispatcher.

For some version-specific processing, the values may be extracted from received messages; for other versions, the values may be determined by algorithm, or by an implementation-defined mechanism. The mechanism by which the value is determined is irrelevant to the Dispatcher.

The following additional or expanded definitions are for use within the Dispatcher.

3.1. messageProcessingModel

The value of messageProcessingModel identifies a Message Processing Model. A Message Processing Model describes the version-specific procedures for extracting data from messages, generating messages, calling upon a securityModel to apply its security services to messages, for converting data from a version-specific message format into a generic format usable by the Dispatcher, and for converting data from Dispatcher format into a version-specific message format.

3.2. pduVersion

The value of pduVersion represents a specific version of protocol operation and its associated PDU formats, such as SNMPv1 or SNMPv2 [RFC3416]. The values of pduVersion are specific to the version of the PDU contained in a message, and the PDUs processed by applications. The Dispatcher does not use the value of pduVersion directly.

An application specifies the pduVersion when it requests the PDU Dispatcher to send a PDU to another SNMP engine. The Dispatcher passes the pduVersion to a Message Processing Model, so it knows how to handle the PDU properly.

For incoming messages, the pduVersion is provided to the Dispatcher by a version-specific Message Processing module. The PDU Dispatcher passes the pduVersion to the application so it knows how to handle the PDU properly. For example, a command responder application needs to know whether to use [RFC3416] elements of procedure and syntax instead of those specified for SNMPv1.

3.3. pduType

A value of the pduType represents a specific type of protocol operation. The values of the pduType are specific to the version of the PDU contained in a message.

Applications register to support particular pduTypes for particular contextEngineIDs.

For incoming messages, pduType is provided to the Dispatcher by a version-specific Message Processing module. It is subsequently used to dispatch the PDU to the application which registered for the pduType for the contextEngineID of the associated scopedPDU.

3.4. sendPduHandle

This handle is generated for coordinating the processing of requests and responses between the SNMP engine and an application. The handle must be unique across all version-specific Message Processing Models, and is of local significance only.

4. Dispatcher Elements of Procedure

This section describes the procedures followed by the Dispatcher when generating and processing SNMP messages.

4.1. Sending an SNMP Message to the Network

This section describes the procedure followed by an SNMP engine whenever it sends an SNMP message.

4.1.1. Sending a Request or Notification

The following procedures are followed by the Dispatcher when an application wants to send an SNMP PDU to another (remote) application, i.e., to initiate a communication by originating a message, such as one containing a request or a notification.

1) The application requests this using the abstract service primitive:

```
statusInformation =              -- sendPduHandle if success
                                 -- errorIndication if failure
  sendPdu(
  IN   transportDomain           -- transport domain to be used
  IN   transportAddress          -- destination network address
  IN   messageProcessingModel    -- typically, SNMP version
  IN   securityModel             -- Security Model to use
  IN   securityName              -- on behalf of this principal
  IN   securityLevel             -- Level of Security requested
  IN   contextEngineID           -- data from/at this entity
  IN   contextName               -- data from/in this context
  IN   pduVersion                -- the version of the PDU
  IN   PDU                       -- SNMP Protocol Data Unit
  IN   expectResponse            -- TRUE or FALSE
       )
```

2) If the messageProcessingModel value does not represent a Message Processing Model known to the Dispatcher, then an errorIndication (implementation-dependent) is returned to the calling application. No further processing is performed.

3) The Dispatcher generates a sendPduHandle to coordinate subsequent processing.

4) The Message Dispatcher sends the request to the version-specific
   Message Processing module identified by messageProcessingModel
   using the abstract service primitive:

```
statusInformation =              -- success or error indication
  prepareOutgoingMessage(
  IN   transportDomain           -- as specified by application
  IN   transportAddress          -- as specified by application
  IN   messageProcessingModel    -- as specified by application
  IN   securityModel             -- as specified by application
  IN   securityName              -- as specified by application
  IN   securityLevel             -- as specified by application
  IN   contextEngineID           -- as specified by application
  IN   contextName               -- as specified by application
  IN   pduVersion                -- as specified by application
  IN   PDU                       -- as specified by application
  IN   expectResponse            -- as specified by application
  IN   sendPduHandle             -- as determined in step 3.
  OUT  destTransportDomain       -- destination transport domain
  OUT  destTransportAddress      -- destination transport address
  OUT  outgoingMessage           -- the message to send
  OUT  outgoingMessageLength     -- the message length
       )
```

5) If the statusInformation indicates an error, the errorIndication
   is returned to the calling application. No further processing is
   performed.

6) If the statusInformation indicates success, the sendPduHandle is

returned to the application, and the outgoingMessage is sent. The transport used to send the outgoingMessage is returned via destTransportDomain, and the address to which it was sent is returned via destTransportAddress.

Outgoing Message Processing is complete.

4.1.2. Sending a Response to the Network

The following procedure is followed when an application wants to return a response back to the originator of an SNMP Request.

1) An application can request this using the abstract service primitive:

```
result =
returnResponsePdu(
 IN   messageProcessingModel   -- typically, SNMP version
 IN   securityModel            -- Security Model in use
 IN   securityName             -- on behalf of this principal
 IN   securityLevel            -- same as on incoming request
 IN   contextEngineID          -- data from/at this SNMP entity
 IN   contextName              -- data from/in this context
 IN   pduVersion               -- the version of the PDU
 IN   PDU                      -- SNMP Protocol Data Unit
 IN   maxSizeResponseScopedPDU -- maximum size of Response PDU
 IN   stateReference           -- reference to state information
                               -- as presented with the request
 IN   statusInformation        -- success or errorIndication
 )                             -- (error counter OID and value
                               -- when errorIndication)
```

2) The Message Dispatcher sends the request to the appropriate

Message Processing Model indicated by the received value of
messageProcessingModel using the abstract service primitive:

```
result =                     -- SUCCESS or errorIndication
 prepareResponseMessage(
 IN   messageProcessingModel   -- specified by application
 IN   securityModel            -- specified by application
 IN   securityName             -- specified by application
 IN   securityLevel            -- specified by application
 IN   contextEngineID          -- specified by application
 IN   contextName              -- specified by application
 IN   pduVersion               -- specified by application
 IN   PDU                      -- specified by application
 IN   maxSizeResponseScopedPDU -- specified by application
 IN   stateReference           -- specified by application
 IN   statusInformation        -- specified by application
 OUT  destTransportDomain      -- destination transport domain
 OUT  destTransportAddress     -- destination transport address
 OUT  outgoingMessage          -- the message to send
 OUT  outgoingMessageLength    -- the message length
     )
```

3) If the result is an errorIndication, the errorIndication is returned to the calling application. No further processing is performed.

4) If the result is success, the outgoingMessage is sent. The transport used to send the outgoingMessage is returned via destTransportDomain, and the address to which it was sent is returned via destTransportAddress.

Message Processing is complete.

4.2. Receiving an SNMP Message from the Network

This section describes the procedure followed by an SNMP engine whenever it receives an SNMP message.

Please note, that for the sake of clarity and to prevent the text from being even longer and more complicated, some details were omitted from the steps below. In particular, the elements of procedure do not always explicitly indicate when state information needs to be released. The general rule is that if state information is available when a message is to be "discarded without further processing", then the state information must also be released at that same time.

4.2.1. Message Dispatching of received SNMP Messages

1) The snmpInPkts counter [RFC3418] is incremented.

2) The version of the SNMP message is determined in an implementation-dependent manner. If the packet cannot be sufficiently parsed to determine the version of the SNMP message, then the snmpInASNParseErrs [RFC3418] counter is incremented, and the message is discarded without further processing. If the version is not supported, then the snmpInBadVersions [RFC3418] counter is incremented, and the message is discarded without further processing.

3) The origin transportDomain and origin transportAddress are determined.

4) The message is passed to the version-specific Message Processing Model which returns the abstract data elements required by the Dispatcher. This is performed using the abstract service primitive:

```
result =                    -- SUCCESS or errorIndication
  prepareDataElements(
  IN   transportDomain        -- origin as determined in step 3.
  IN   transportAddress       -- origin as determined in step 3.
  IN   wholeMsg               -- as received from the network
  IN   wholeMsgLength         -- as received from the network
  OUT  messageProcessingModel -- typically, SNMP version
  OUT  securityModel          -- Security Model specified
  OUT  securityName           -- on behalf of this principal
  OUT  securityLevel          -- Level of Security specified
  OUT  contextEngineID        -- data from/at this entity
  OUT  contextName            -- data from/in this context
  OUT  pduVersion             -- the version of the PDU
  OUT  PDU                    -- SNMP Protocol Data Unit
  OUT  pduType                -- SNMP PDU type
  OUT  sendPduHandle          -- handle for a matched request
  OUT  maxSizeResponseScopedPDU -- maximum size of Response PDU
  OUT  statusInformation      -- success or errorIndication
                              -- (error counter OID and value
                              -- when errorIndication)
  OUT  stateReference         -- reference to state information
                              -- to be used for a possible
     )                        -- Response
```

5) If the result is a FAILURE errorIndication, the message is discarded without further processing.

6) At this point, the abstract data elements have been prepared and processing continues as described in Section 4.2.2, PDU Dispatching for Incoming Messages.

4.2.2. PDU Dispatching for Incoming Messages

The elements of procedure for the dispatching of PDUs depends on the value of sendPduHandle. If the value of sendPduHandle is <none>, then this is a request or notification and the procedures specified in Section 4.2.2.1 apply. If the value of snmpPduHandle is not <none>, then this is a response and the procedures specified in Section 4.2.2.2 apply.

4.2.2.1. Incoming Requests and Notifications

The following procedures are followed for the dispatching of PDUs when the value of sendPduHandle is <none>, indicating this is a request or notification.

1) The combination of contextEngineID and pduType is used to determine which application has registered for this request or notification.

2) If no application has registered for the combination, then:

a) The snmpUnknownPDUHandlers counter is incremented.

b) A Response message is generated using the abstract service primitive:

```
result =                    -- SUCCESS or FAILURE
prepareResponseMessage(
IN   messageProcessingModel    -- as provided by MP module
IN   securityModel             -- as provided by MP module
IN   securityName              -- as provided by MP module
IN   securityLevel             -- as provided by MP module
IN   contextEngineID           -- as provided by MP module
IN   contextName               -- as provided by MP module
IN   pduVersion                -- as provided by MP module
IN   PDU                       -- as provided by MP module
IN   maxSizeResponseScopedPDU  -- as provided by MP module
IN   stateReference            -- as provided by MP module
IN   statusInformation         -- errorIndication plus
                               -- snmpUnknownPDUHandlers OID
                               -- value pair.
OUT  destTransportDomain       -- destination transportDomain
OUT  destTransportAddress      -- destination transportAddress
OUT  outgoingMessage           -- the message to send
OUT  outgoingMessageLength     -- its length
)
```

c) If the result is SUCCESS, then the prepared message is sent to the originator of the request as identified by the transportDomain and transportAddress. The transport used to send the outgoingMessage is returned via destTransportDomain, and the address to which it was sent is returned via destTransportAddress.

d) The incoming message is discarded without further processing. Message Processing for this message is complete.

3) The PDU is dispatched to the application, using the abstract service primitive:

```
processPdu(                    -- process Request/Notification
  IN   messageProcessingModel   -- as provided by MP module
  IN   securityModel            -- as provided by MP module
  IN   securityName             -- as provided by MP module
  IN   securityLevel            -- as provided by MP module
  IN   contextEngineID          -- as provided by MP module
  IN   contextName              -- as provided by MP module
  IN   pduVersion               -- as provided by MP module
  IN   PDU                      -- as provided by MP module
  IN   maxSizeResponseScopedPDU -- as provided by MP module
  IN   stateReference           -- as provided by MP module
                                -- needed when sending response
     )
```

Message processing for this message is complete.

4.2.2.2. Incoming Responses

The following procedures are followed for the dispatching of PDUs when the value of sendPduHandle is not <none>, indicating this is a response.

1) The value of sendPduHandle is used to determine, in an implementation-defined manner, which application is waiting for a response associated with this sendPduHandle.

2) If no waiting application is found, the message is discarded without further processing, and the stateReference is released. The snmpUnknownPDUHandlers counter is incremented. Message Processing is complete for this message.

3) Any cached information, including stateReference, about the message is discarded.

4) The response is dispatched to the application using the abstract
   service primitive:

```
processResponsePdu(              -- process Response PDU
  IN   messageProcessingModel    -- provided by the MP module
  IN   securityModel             -- provided by the MP module
  IN   securityName              -- provided by the MP module
  IN   securityLevel             -- provided by the MP module
  IN   contextEngineID           -- provided by the MP module
  IN   contextName               -- provided by the MP module
  IN   pduVersion                -- provided by the MP module
  IN   PDU                       -- provided by the MP module
  IN   statusInformation         -- provided by the MP module
  IN   sendPduHandle             -- provided by the MP module
       )
```

   Message Processing is complete for this message.

4.3. Application Registration for Handling PDU types

Applications that want to process certain PDUs must register with the PDU Dispatcher. Applications specify the combination of contextEngineID and pduType(s) for which they want to take responsibility.

1) An application registers according to the abstract interface
   primitive:

```
statusInformation =           -- success or errorIndication
```

```
registerContextEngineID(
IN   contextEngineID       -- take responsibility for this one
IN   pduType               -- the pduType(s) to be registered
    )
```

Note: Implementations may provide a means of requesting registration for simultaneous multiple contextEngineID values, e.g., all contextEngineID values, and may also provide a means for requesting simultaneous registration for multiple values of the pduType.

2) The parameters may be checked for validity; if they are not, then an errorIndication (invalidParameter) is returned to the application.

3) Each combination of contextEngineID and pduType can be registered only once. If another application has already registered for the specified combination, then an errorIndication (alreadyRegistered) is returned to the application.

4) Otherwise, the registration is saved so that SNMP PDUs can be dispatched to this application.

4.4. Application Unregistration for Handling PDU Types

Applications that no longer want to process certain PDUs must unregister with the PDU Dispatcher.

1) An application unregisters using the abstract service primitive:

```
unregisterContextEngineID(
 IN   contextEngineID       -- give up responsibility for this
 IN   pduType               -- the pduType(s) to be unregistered
     )
```

Note: Implementations may provide a means for requesting the unregistration for simultaneous multiple contextEngineID values, e.g., all contextEngineID values, and may also provide a means for requesting simultaneous unregistration for multiple values of pduType.

2) If the contextEngineID and pduType combination has been
   registered, then the registration is deleted.

   If no such registration exists, then the request is ignored.

5. Definitions

5.1. Definitions for SNMP Message Processing and Dispatching

```
SNMP-MPD-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-COMPLIANCE, OBJECT-GROUP         FROM SNMPv2-CONF
    MODULE-IDENTITY, OBJECT-TYPE,
    snmpModules, Counter32                  FROM SNMPv2-SMI;

snmpMPDMIB MODULE-IDENTITY
    LAST-UPDATED "200210140000Z"
    ORGANIZATION "SNMPv3 Working Group"
    CONTACT-INFO "WG-EMail:   snmpv3@lists.tislabs.com
                  Subscribe:  snmpv3-request@lists.tislabs.com

                  Co-Chair:   Russ Mundy
                              Network Associates Laboratories
                  postal:     15204 Omega Drive, Suite 300
                              Rockville, MD 20850-4601
                              USA
```



```
                  EMail:      mundy@tislabs.com
                  phone:      +1 301-947-7107

                  Co-Chair &
                  Co-editor:  David Harrington
                              Enterasys Networks
                  postal:     35 Industrial Way
                              P. O. Box 5005
                              Rochester NH 03866-5005
                              USA
                  EMail:      dbh@enterasys.com
                  phone:      +1 603-337-2614
```

Co-editor: Jeffrey Case
SNMP Research, Inc.
postal: 3001 Kimberlin Heights Road
Knoxville, TN 37920-9716
USA
EMail: case@snmp.com
phone: +1 423-573-1434

Co-editor: Randy Presuhn
BMC Software, Inc.
postal: 2141 North First Street
San Jose, CA 95131
USA
EMail: randy_presuhn@bmc.com
phone: +1 408-546-1006

Co-editor: Bert Wijnen
Lucent Technologies
postal: Schagen 33
3461 GL Linschoten
Netherlands
EMail: bwijnen@lucent.com
phone: +31 348-680-485
"

DESCRIPTION "The MIB for Message Processing and Dispatching

Copyright (C) The Internet Society (2002). This
version of this MIB module is part of RFC 3412;
see the RFC itself for full legal notices.
"

REVISION "200210140000Z" -- 14 October 2002
DESCRIPTION "Updated addresses, published as RFC 3412."
REVISION "199905041636Z" -- 4 May 1999
DESCRIPTION "Updated addresses, published as RFC 2572."

REVISION "199709300000Z" -- 30 September 1997
DESCRIPTION "Original version, published as RFC 2272."
::= { snmpModules 11 }

-- Administrative assignments ***************************************

```
snmpMPDAdmin           OBJECT IDENTIFIER ::= { snmpMPDMIB 1 }
snmpMPDMIBObjects      OBJECT IDENTIFIER ::= { snmpMPDMIB 2 }
snmpMPDMIBConformance  OBJECT IDENTIFIER ::= { snmpMPDMIB 3 }

-- Statistics for SNMP Messages *************************************

snmpMPDStats           OBJECT IDENTIFIER ::= { snmpMPDMIBObjects 1 }

snmpUnknownSecurityModels OBJECT-TYPE
   SYNTAX       Counter32
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION "The total number of packets received by the SNMP
             engine which were dropped because they referenced a
             securityModel that was not known to or supported by
             the SNMP engine.
            "
   ::= { snmpMPDStats 1 }

snmpInvalidMsgs OBJECT-TYPE
   SYNTAX       Counter32
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION "The total number of packets received by the SNMP
             engine which were dropped because there were invalid
             or inconsistent components in the SNMP message.
            "
   ::= { snmpMPDStats 2 }

snmpUnknownPDUHandlers OBJECT-TYPE
   SYNTAX       Counter32
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION "The total number of packets received by the SNMP
             engine which were dropped because the PDU contained
             in the packet could not be passed to an application
             responsible for handling the pduType, e.g. no SNMP
             application had registered for the proper
             combination of the contextEngineID and the pduType.
            "
   ::= { snmpMPDStats 3 }
```

```
-- Conformance information ******************************************

snmpMPDMIBCompliances OBJECT IDENTIFIER ::= {snmpMPDMIBConformance 1}
snmpMPDMIBGroups      OBJECT IDENTIFIER ::= {snmpMPDMIBConformance 2}

-- Compliance statements

snmpMPDCompliance MODULE-COMPLIANCE
   STATUS      current
   DESCRIPTION "The compliance statement for SNMP entities which
               implement the SNMP-MPD-MIB.
              "
   MODULE    -- this module
      MANDATORY-GROUPS { snmpMPDGroup }
   ::= { snmpMPDMIBCompliances 1 }

snmpMPDGroup OBJECT-GROUP
   OBJECTS {
             snmpUnknownSecurityModels,
             snmpInvalidMsgs,
             snmpUnknownPDUHandlers
           }
   STATUS      current
   DESCRIPTION "A collection of objects providing for remote
               monitoring of the SNMP Message Processing and
               Dispatching process.
              "
   ::= { snmpMPDMIBGroups 1 }

END
```

6. The SNMPv3 Message Format

This section defines the SNMPv3 message format and the corresponding SNMP version 3 Message Processing Model (v3MP).

```
SNMPv3MessageSyntax DEFINITIONS IMPLICIT TAGS ::= BEGIN

   SNMPv3Message ::= SEQUENCE {
      -- identify the layout of the SNMPv3Message
      -- this element is in same position as in SNMPv1
      -- and SNMPv2c, allowing recognition
      -- the value 3 is used for snmpv3
      msgVersion INTEGER ( 0 .. 2147483647 ),
      -- administrative parameters
```

```
    msgGlobalData HeaderData,
    -- security model-specific parameters
    -- format defined by Security Model
```



```
    msgSecurityParameters OCTET STRING,
    msgData  ScopedPduData
  }

  HeaderData ::= SEQUENCE {
    msgID      INTEGER (0..2147483647),
    msgMaxSize INTEGER (484..2147483647),

    msgFlags   OCTET STRING (SIZE(1)),
          --  .... ...1   authFlag
          --  .... ..1.   privFlag
          --  .... .1..   reportableFlag
          --              Please observe:
          --  .... ..00   is OK, means noAuthNoPriv
          --  .... ..01   is OK, means authNoPriv
          --  .... ..10   reserved, MUST NOT be used.
          --  .... ..11   is OK, means authPriv

    msgSecurityModel INTEGER (1..2147483647)
  }

  ScopedPduData ::= CHOICE {
    plaintext    ScopedPDU,
    encryptedPDU OCTET STRING  -- encrypted scopedPDU value
  }

  ScopedPDU ::= SEQUENCE {
    contextEngineID  OCTET STRING,
    contextName      OCTET STRING,
    data             ANY -- e.g., PDUs as defined in [RFC3416]
  }
 END
```

6.1. msgVersion

The msgVersion field is set to snmpv3(3) and identifies the message as an SNMP version 3 Message.

6.2. msgID

The msgID is used between two SNMP entities to coordinate request messages and responses, and by the v3MP to coordinate the processing of the message by different subsystem models within the architecture.

Values for msgID SHOULD be generated in a manner that avoids re-use of any outstanding values. Doing so provides protection against some replay attacks. One possible implementation strategy would be to use the low-order bits of snmpEngineBoots [RFC3411] as the high-order

portion of the msgID value and a monotonically increasing integer for the low-order portion of msgID.

Note that the request-id in a PDU may be used by SNMP applications to identify the PDU; the msgID is used by the engine to identify the message which carries a PDU. The engine needs to identify the message even if decryption of the PDU (and request-id) fails. No assumption should be made that the value of the msgID and the value of the request-id are equivalent.

The value of the msgID field for a response takes the value of the msgID field from the message to which it is a response. By use of the msgID value, an engine can distinguish the (potentially multiple) outstanding requests, and thereby correlate incoming responses with outstanding requests. In cases where an unreliable datagram service is used, the msgID also provides a simple means of identifying messages duplicated by the network. If a request is retransmitted, a new msgID value SHOULD be used for each retransmission.

6.3. msgMaxSize

The msgMaxSize field of the message conveys the maximum message size supported by the sender of the message, i.e., the maximum message size that the sender can accept when another SNMP engine sends an SNMP message (be it a response or any other message) to the sender of this message on the transport in use for this message.

When an SNMP message is being generated, the msgMaxSize is provided by the SNMP engine which generates the message. At the receiving

SNMP engine, the msgMaxSize is used to determine the maximum message size the sender can accommodate.

6.4. msgFlags

The msgFlags field of the message contains several bit fields which control processing of the message.

The reportableFlag is a secondary aid in determining whether a Report PDU MUST be sent. It is only used in cases where the PDU portion of a message cannot be decoded, due to, for example, an incorrect encryption key. If the PDU can be decoded, the PDU type forms the basis for decisions on sending Report PDUs.

When the reportableFlag is used, if its value is one, a Report PDU MUST be returned to the sender under those conditions which can cause the generation of Report PDUs. Similarly, when the reportableFlag is used and its value is zero, then a Report PDU MUST NOT be sent. The reportableFlag MUST always be zero when the message contains a PDU

from the Unconfirmed Class, such as a Report PDU, a response-type PDU (such as a Response PDU), or an unacknowledged notification-type PDU (such as an SNMPv2-trap PDU). The reportableFlag MUST always be one for a PDU from the Confirmed Class, including request-type PDUs (such as a Get PDU) and acknowledged notification-type PDUs (such as an Inform PDU).

If the reportableFlag is set to one for a message containing a PDU from the Unconfirmed Class, such as a Report PDU, a response-type PDU (such as a Response PDU), or an unacknowledged notification-type PDU (such as an SNMPv2-trap PDU), then the receiver of that message MUST process it as though the reportableFlag had been set to zero.

If the reportableFlag is set to zero for a message containing a request-type PDU (such as a Get PDU) or an acknowledged notification-type PDU (such as an Inform PDU), then the receiver of that message MUST process it as though the reportableFlag had been set to one.

Report PDUs are generated directly by the SNMPv3 Message Processing Model, and support engine-to-engine communications, but may be passed to applications for processing.

An SNMP engine that receives a reportPDU may use it to determine what kind of problem was detected by the remote SNMP engine. It can do so based on the error counter included as the first (and only) varBind of the reportPDU. Based on the detected error, the SNMP engine may try to send a corrected SNMP message. If that is not possible, it may pass an indication of the error to the application on whose behalf the failed SNMP request was issued.

The authFlag and privFlag portions of the msgFlags field are set by the sender to indicate the securityLevel that was applied to the message before it was sent on the wire. The receiver of the message MUST apply the same securityLevel when the message is received and the contents are being processed.

There are three securityLevels, namely noAuthNoPriv, which is less than authNoPriv, which is in turn less than authPriv. See the SNMP architecture document [RFC3411] for details about the securityLevel.

a) authFlag

If the authFlag is set to one, then the securityModel used by the SNMP engine which sent the message MUST identify the securityName on whose behalf the SNMP message was generated and MUST provide, in a securityModel-specific manner, sufficient data for the receiver of the message to be able to authenticate that

identification. In general, this authentication will allow the receiver to determine with reasonable certainty that the message was:

- sent on behalf of the principal associated with the securityName,

- was not redirected,

- was not modified in transit, and

- was not replayed.

If the authFlag is zero, then the securityModel used by the SNMP engine which sent the message MUST identify the securityName on whose behalf the SNMP message was generated but it does not need to provide sufficient data for the receiver of the message to authenticate the identification, as there is no need to authenticate the message in this case.

b) privFlag

If the privFlag is set, then the securityModel used by the SNMP engine which sent the message MUST also protect the scopedPDU in an SNMP message from disclosure, i.e., it MUST encrypt/decrypt the scopedPDU. If the privFlag is zero, then the securityModel in use does not need to protect the data from disclosure.

It is an explicit requirement of the SNMP architecture that if privacy is selected, then authentication is also required. That means that if the privFlag is set, then the authFlag MUST also be set to one.

The combination of the authFlag and the privFlag comprises a Level of Security as follows:

```
authFlag zero, privFlag zero -> securityLevel is noAuthNoPriv
authFlag zero, privFlag one  -> invalid combination, see below
authFlag one,  privFlag zero -> securityLevel is authNoPriv
authFlag one,  privFlag one  -> securityLevel is authPriv
```

The elements of procedure (see below) describe the action to be taken when the invalid combination of authFlag equal to zero and privFlag equal to one is encountered.

The remaining bits in msgFlags are reserved, and MUST be set to zero when sending a message and SHOULD be ignored when receiving a message.

6.5. msgSecurityModel

The v3MP supports the concurrent existence of multiple Security Models to provide security services for SNMPv3 messages. The msgSecurityModel field in an SNMPv3 Message identifies which Security

Model was used by the sender to generate the message and therefore which securityModel MUST be used by the receiver to perform security processing for the message. The mapping to the appropriate securityModel implementation within an SNMP engine is accomplished in an implementation-dependent manner.

6.6. msgSecurityParameters

The msgSecurityParameters field of the SNMPv3 Message is used for communication between the Security Model modules in the sending and receiving SNMP engines. The data in the msgSecurityParameters field is used exclusively by the Security Model, and the contents and format of the data is defined by the Security Model. This OCTET STRING is not interpreted by the v3MP, but is passed to the local implementation of the Security Model indicated by the msgSecurityModel field in the message.

6.7. scopedPduData

The scopedPduData field represents either the plain text scopedPDU if the privFlag in the msgFlags is zero, or it represents an encryptedPDU (encoded as an OCTET STRING) which MUST be decrypted by the securityModel in use to produce a plaintext scopedPDU.

6.8. scopedPDU

The scopedPDU contains information to identify an administratively unique context and a PDU. The object identifiers in the PDU refer to managed objects which are (expected to be) accessible within the specified context.

6.8.1. contextEngineID

The contextEngineID in the SNMPv3 message uniquely identifies, within an administrative domain, an SNMP entity that may realize an instance of a context with a particular contextName.

For incoming messages, the contextEngineID is used in conjunction with the pduType to determine to which application the scopedPDU will be sent for processing.

For outgoing messages, the v3MP sets the contextEngineID to the value provided by the application in the request for a message to be sent.

6.8.2. contextName

The contextName field in an SNMPv3 message, in conjunction with the contextEngineID field, identifies the particular context associated with the management information contained in the PDU portion of the message. The contextName is unique within the SNMP entity specified by the contextEngineID, which may realize the managed objects referenced within the PDU. An application which originates a message provides the value for the contextName field and this value may be used during processing by an application at the receiving SNMP Engine.

6.8.3. data

The data field of the SNMPv3 Message contains the PDU. Among other things, the PDU contains the PDU type that is used by the v3MP to determine the type of the incoming SNMP message. The v3MP specifies that the PDU MUST be one of those specified in [RFC3416].

7. Elements of Procedure for v3MP

This section describes the procedures followed by an SNMP engine when generating and processing SNMP messages according to the SNMPv3 Message Processing Model.

Please note, that for the sake of clarity and to prevent the text from being even longer and more complicated, some details were omitted from the steps below.

a) Some steps specify that when some error conditions are encountered when processing a received message, a message containing a Report PDU is generated and the received message is discarded without further processing. However, a Report-PDU MUST NOT be generated unless the PDU causing generation of the Report PDU can be determined to be a member of the Confirmed Class, or the reportableFlag is set to one and the PDU class cannot be determined.

b) The elements of procedure do not always explicitly indicate when state information needs to be released. The general rule is that if state information is available when a message is to be "discarded without further processing", then the state information should also be released at that same time.

7.1. Prepare an Outgoing SNMP Message

This section describes the procedure followed to prepare an SNMPv3 message from the data elements passed by the Message Dispatcher.

1) The Message Dispatcher may request that an SNMPv3 message containing a Read Class, Write Class, or Notification Class PDU be prepared for sending.

a) It makes such a request according to the abstract service primitive:

```
statusInformation =           -- success or errorIndication
  prepareOutgoingMessage(
  IN   transportDomain        -- requested transport domain
  IN   transportAddress       -- requested destination address
  IN   messageProcessingModel -- typically, SNMP version
  IN   securityModel          -- Security Model to use
  IN   securityName           -- on behalf of this principal
  IN   securityLevel          -- Level of Security requested
  IN   contextEngineID        -- data from/at this entity
  IN   contextName            -- data from/in this context
  IN   pduVersion             -- version of the PDU *
  IN   PDU                    -- SNMP Protocol Data Unit
  IN   expectResponse         -- TRUE or FALSE *
  IN   sendPduHandle          -- the handle for matching
                              -- incoming responses
  OUT  destTransportDomain    -- destination transport domain
  OUT  destTransportAddress   -- destination transport address
  OUT  outgoingMessage        -- the message to send
  OUT  outgoingMessageLength  -- the length of the message
  )
```

* The SNMPv3 Message Processing Model does not use the values of expectResponse or pduVersion.

b) A unique msgID is generated. The number used for msgID should not have been used recently, and MUST NOT be the same as was used for any outstanding request.

2) The Message Dispatcher may request that an SNMPv3 message containing a Response Class or Internal Class PDU be prepared for sending.

a) It makes such a request according to the abstract service primitive:

```
result =                       -- SUCCESS or FAILURE
prepareResponseMessage(
 IN   messageProcessingModel   -- typically, SNMP version
 IN   securityModel            -- same as on incoming request
 IN   securityName             -- same as on incoming request
 IN   securityLevel            -- same as on incoming request
 IN   contextEngineID          -- data from/at this SNMP entity
 IN   contextName              -- data from/in this context
 IN   pduVersion               -- version of the PDU
 IN   PDU                      -- SNMP Protocol Data Unit
 IN   maxSizeResponseScopedPDU -- maximum size sender can
                               -- accept
 IN   stateReference           -- reference to state
                               -- information presented with
                               -- the request
 IN   statusInformation        -- success or errorIndication
                               -- error counter OID and value
                               -- when errorIndication
 OUT  destTransportDomain      -- destination transport domain
 OUT  destTransportAddress     -- destination transport address
 OUT  outgoingMessage          -- the message to send
 OUT  outgoingMessageLength    -- the length of the message
 )
```

b) The cached information for the original request is retrieved via the stateReference, including:

- msgID,
- contextEngineID,
- contextName,
- securityModel,
- securityName,
- securityLevel,
- securityStateReference,
- reportableFlag,
- transportDomain, and
- transportAddress.

The SNMPv3 Message Processing Model does not allow cached data to be overridden, except by error indications as detailed in (3) below.

3) If statusInformation contains values for an OID/value combination (potentially also containing a securityLevel value, contextEngineID value, or contextName value), then:

   a) If a PDU is provided, it is the PDU from the original request. If possible, extract the request-id and pduType.

   b) If the pduType is determined to not be a member of the Confirmed Class, or if the reportableFlag is zero and the pduType cannot be determined, then the original message is discarded, and no further processing is done. A result of FAILURE is returned. SNMPv3 Message Processing is complete.

   c) A Report PDU is prepared:

      1) the varBindList is set to contain the OID and value from the statusInformation.

      2) error-status is set to 0.

      3) error-index is set to 0.

4) request-id is set to the value extracted in step b).
Otherwise, request-id is set to 0.

d) The errorIndication in statusInformation may be accompanied by a securityLevel value, a contextEngineID value, or a contextName value.

1) If statusInformation contains a value for securityLevel, then securityLevel is set to that value, otherwise it is set to noAuthNoPriv.

2) If statusInformation contains a value for contextEngineID, then contextEngineID is set to that value, otherwise it is set to the value of this entity's snmpEngineID.

3) If statusInformation contains a value for contextName, then contextName is set to that value, otherwise it is set to the default context of "" (zero-length string).

e) PDU is set to refer to the new Report-PDU. The old PDU is discarded.

f) Processing continues with step 6) below.

4) If the contextEngineID is not yet determined, then the contextEngineID is determined, in an implementation-dependent manner, possibly using the transportDomain and transportAddress.

5) If the contextName is not yet determined, the contextName is set to the default context.

6) A scopedPDU is prepared from the contextEngineID, contextName, and PDU.

7) msgGlobalData is constructed as follows:

a) The msgVersion field is set to snmpv3(3).

b) msgID is set as determined in step 1 or 2 above.

c) msgMaxSize is set to an implementation-dependent value.

d) msgFlags are set as follows:

- If securityLevel specifies noAuthNoPriv, then authFlag and privFlag are both set to zero.

- If securityLevel specifies authNoPriv, then authFlag is set to one and privFlag is set to zero.

- If securityLevel specifies authPriv, then authFlag is set to one and privFlag is set to one.

- If the PDU is from the Unconfirmed Class, then the reportableFlag is set to zero.

- If the PDU is from the Confirmed Class then the reportableFlag is set to one.

- All other msgFlags bits are set to zero.

e) msgSecurityModel is set to the value of securityModel.

8) If the PDU is from the Response Class or the Internal Class, then:

a) The specified Security Model is called to generate the message according to the primitive:

```
statusInformation =
  generateResponseMsg(
  IN   messageProcessingModel -- SNMPv3 Message Processing
                              -- Model
  IN   globalData             -- msgGlobalData from step 7
  IN   maxMessageSize         -- from msgMaxSize (step 7c)
  IN   securityModel          -- as determined in step 7e
  IN   securityEngineID       -- the value of snmpEngineID
  IN   securityName           -- on behalf of this principal
  IN   securityLevel          -- for the outgoing message
  IN   scopedPDU              -- as prepared in step 6)
  IN   securityStateReference -- as determined in step 2
  OUT  securityParameters     -- filled in by Security Module
  OUT  wholeMsg               -- complete generated message
  OUT  wholeMsgLength         -- length of generated message
  )
```

If, upon return from the Security Model, the statusInformation includes an errorIndication, then any cached information about the outstanding request message is discarded, and an errorIndication is returned, so it can be returned to the calling application. SNMPv3 Message Processing is complete.

b) A SUCCESS result is returned. SNMPv3 Message Processing is complete.

9) If the PDU is from the Confirmed Class or the Notification Class, then:

a) If the PDU is from the Unconfirmed Class, then securityEngineID is set to the value of this entity's snmpEngineID.

Otherwise, the snmpEngineID of the target entity is determined, in an implementation-dependent manner, possibly using transportDomain and transportAddress. The value of the securityEngineID is set to the value of the target entity's snmpEngineID.

b) The specified Security Model is called to generate the message
according to the primitive:

```
statusInformation =
 generateRequestMsg(
 IN  messageProcessingModel -- SNMPv3 Message Processing Model
 IN  globalData             -- msgGlobalData, from step 7
 IN  maxMessageSize         -- from msgMaxSize in step 7 c)
 IN  securityModel          -- as provided by caller
 IN  securityEngineID       -- authoritative SNMP entity
                            -- from step 9 a)
 IN  securityName           -- as provided by caller
 IN  securityLevel          -- as provided by caller
 IN  scopedPDU              -- as prepared in step 6
 OUT securityParameters     -- filled in by Security Module
 OUT wholeMsg               -- complete generated message
 OUT wholeMsgLength         -- length of the generated message
 )
```

If, upon return from the Security Model, the statusInformation
includes an errorIndication, then the message is discarded, and
the errorIndication is returned, so it can be returned to the
calling application, and no further processing is done. SNMPv3
Message Processing is complete.

c) If the PDU is from the Confirmed Class, information about the
outgoing message is cached, and an implementation-specific
stateReference is created. Information to be cached includes
the values of:

- sendPduHandle
- msgID
- snmpEngineID
- securityModel
- securityName
- securityLevel
- contextEngineID
- contextName

d) A SUCCESS result is returned. SNMPv3 Message Processing is
complete.

7.2. Prepare Data Elements from an Incoming SNMP Message

This section describes the procedure followed to extract data from an SNMPv3 message, and to prepare the data elements required for further processing of the message by the Message Dispatcher.

1) The message is passed in from the Message Dispatcher according to the abstract service primitive:

```
result =                       -- SUCCESS or errorIndication
  prepareDataElements(
  IN  transportDomain          -- origin transport domain
  IN  transportAddress         -- origin transport address
  IN  wholeMsg                 -- as received from the network
  IN  wholeMsgLength           -- as received from the network
  OUT messageProcessingModel   -- typically, SNMP version
  OUT securityModel            -- Security Model to use
  OUT securityName             -- on behalf of this principal
  OUT securityLevel            -- Level of Security requested
  OUT contextEngineID          -- data from/at this entity
  OUT contextName              -- data from/in this context
  OUT pduVersion               -- version of the PDU
  OUT PDU                      -- SNMP Protocol Data Unit
  OUT pduType                  -- SNMP PDU type
  OUT sendPduHandle            -- handle for matched request
  OUT maxSizeResponseScopedPDU -- maximum size sender can accept
  OUT statusInformation        -- success or errorIndication
                               -- error counter OID and value
                               -- when errorIndication
  OUT stateReference           -- reference to state information
                               -- to be used for a possible
  )                            -- Response
```

2) If the received message is not the serialization (according to the conventions of [RFC3417]) of an SNMPv3Message value, then the snmpInASNParseErrs counter [RFC3418] is incremented, the message is discarded without further processing, and a FAILURE result is

returned. SNMPv3 Message Processing is complete.

3) The values for msgVersion, msgID, msgMaxSize, msgFlags, msgSecurityModel, msgSecurityParameters, and msgData are extracted from the message.

4) If the value of the msgSecurityModel component does not match a supported securityModel, then the snmpUnknownSecurityModels counter is incremented, the message is discarded without further processing, and a FAILURE result is returned. SNMPv3 Message Processing is complete.

5) The securityLevel is determined from the authFlag and the privFlag bits of the msgFlags component as follows:

a) If the authFlag is not set and the privFlag is not set, then securityLevel is set to noAuthNoPriv.

b) If the authFlag is set and the privFlag is not set, then securityLevel is set to authNoPriv.

c) If the authFlag is set and the privFlag is set, then securityLevel is set to authPriv.

d) If the authFlag is not set and privFlag is set, then the snmpInvalidMsgs counter is incremented, the message is discarded without further processing, and a FAILURE result is returned. SNMPv3 Message Processing is complete.

e) Any other bits in the msgFlags are ignored.

6) The security module implementing the Security Model as specified by the securityModel component is called for authentication and privacy services. This is done according to the abstract service primitive:

```
statusInformation =            -- errorIndication or success
                               -- error counter OID and
                               -- value if error
  processIncomingMsg(
  IN  messageProcessingModel   -- SNMPv3 Message Processing Model
```

```
IN  maxMessageSize          -- of the sending SNMP entity
IN  securityParameters      -- for the received message
IN  securityModel           -- for the received message
IN  securityLevel           -- Level of Security
IN  wholeMsg                -- as received on the wire
IN  wholeMsgLength          -- length as received on the wire
OUT securityEngineID        -- authoritative SNMP entity
OUT securityName            -- identification of the principal
OUT scopedPDU,              -- message (plaintext) payload
OUT maxSizeResponseScopedPDU -- maximum size sender can accept
OUT securityStateReference   -- reference to security state
)                           -- information, needed for
                            -- response
```

If an errorIndication is returned by the security module, then:

a) If statusInformation contains values for an OID/value pair, then generation of a Report PDU is attempted (see step 3 in section 7.1).

1) If the scopedPDU has been returned from processIncomingMsg, then determine contextEngineID, contextName, and PDU.

2) Information about the message is cached and a stateReference is created (implementation-specific). Information to be cached includes the values of:

```
msgVersion,
msgID,
securityLevel,
msgFlags,
msgMaxSize,
securityModel,
maxSizeResponseScopedPDU,
securityStateReference
```

3) Request that a Report-PDU be prepared and sent, according to the abstract service primitive:

```
result =                    -- SUCCESS or FAILURE
returnResponsePdu(
```

```
          IN  messageProcessingModel   -- SNMPv3(3)
          IN  securityModel            -- same as on incoming request
          IN  securityName             -- from processIncomingMsg
          IN  securityLevel            -- same as on incoming request
          IN  contextEngineID          -- from step 6 a) 1)
          IN  contextName              -- from step 6 a) 1)
          IN  pduVersion               -- SNMPv2-PDU
          IN  PDU                      -- from step 6 a) 1)
          IN  maxSizeResponseScopedPDU -- from processIncomingMsg
          IN  stateReference           -- from step 6 a) 2)
          IN  statusInformation        -- from processIncomingMsg
          )
```

b) The incoming message is discarded without further processing, and a FAILURE result is returned. SNMPv3 Message Processing is complete.

7) The scopedPDU is parsed to extract the contextEngineID, the contextName and the PDU. If any parse error occurs, then the snmpInASNParseErrs counter [RFC3418] is incremented, the security state information is discarded, the message is discarded without further processing, and a FAILURE result is returned. SNMPv3 Message Processing is complete. Treating an unknown PDU type is treated as a parse error is an implementation option.

8) The pduVersion is determined in an implementation-dependent manner. For SNMPv3, the pduVersion would be an SNMPv2-PDU.

9) The pduType is determined, in an implementation-dependent manner. For [RFC3416], the pduTypes include:

- GetRequest-PDU,
- GetNextRequest-PDU,
- GetBulkRequest-PDU,
- SetRequest-PDU,
- InformRequest-PDU,
- SNMPv2-Trap-PDU,
- Response-PDU,

- Report-PDU.

10) If the pduType is from the Response Class or the Internal Class, then:

a) The value of the msgID component is used to find the cached information for a corresponding outstanding Request message. If no such outstanding Request message is found, then the security state information is discarded, the message is discarded without further processing, and a FAILURE result is returned. SNMPv3 Message Processing is complete.

b) sendPduHandle is retrieved from the cached information.

Otherwise, sendPduHandle is set to <none>, an implementation defined value.

11) If the pduType is from the Internal Class, then:

a) statusInformation is created using the contents of the Report-PDU, in an implementation-dependent manner. This statusInformation will be forwarded to the application associated with the sendPduHandle.

b) The cached data for the outstanding message, referred to by stateReference, is retrieved. If the securityModel or securityLevel values differ from the cached ones, it is important to recognize that Internal Class PDUs delivered at the security level of noAuthNoPriv open a window of opportunity for spoofing or replay attacks. If the receiver of such messages is aware of these risks, the use of such unauthenticated messages is acceptable and may provide a useful function for discovering engine IDs or for detecting misconfiguration at remote nodes.

When the securityModel or securityLevel values differ from the cached ones, an implementation may retain the cached information about the outstanding Request message, in anticipation of the possibility that the Internal Class PDU received might be illegitimate. Otherwise, any cached information about the outstanding Request message is discarded.

c) The security state information for this incoming message is discarded.

d) stateReference is set to <none>.

e) A SUCCESS result is returned. SNMPv3 Message Processing is complete.

12) If the pduType is from the Response Class, then:

a) The cached data for the outstanding request, referred to by stateReference, is retrieved, including:

- snmpEngineID
- securityModel
- securityName
- securityLevel
- contextEngineID
- contextName

b) If the values extracted from the incoming message differ from the cached data, then any cached information about the outstanding Request message is discarded, the incoming message is discarded without further processing, and a FAILURE result is returned. SNMPv3 Message Processing is complete.

When the securityModel or securityLevel values differ from the cached ones, an implementation may retain the cached information about the outstanding Request message, in anticipation of the possibility that the Response Class PDU received might be illegitimate.

c) Otherwise, any cached information about the outstanding Request message is discarded, and the stateReference is set to <none>.

d) A SUCCESS result is returned. SNMPv3 Message Processing is complete.

13) If the pduType is from the Confirmed Class, then:

a) If the value of securityEngineID is not equal to the value of snmpEngineID, then the security state information is discarded, any cached information about this message is discarded, the incoming message is discarded without further processing, and a FAILURE result is returned. SNMPv3 Message Processing is complete.

b) Information about the message is cached and a stateReference is created (implementation-specific). Information to be cached includes the values of:

msgVersion,
msgID,
securityLevel,
msgFlags,
msgMaxSize,
securityModel,
maxSizeResponseScopedPDU,
securityStateReference

c) A SUCCESS result is returned. SNMPv3 Message Processing is complete.

14) If the pduType is from the Unconfirmed Class, then a SUCCESS result is returned. SNMPv3 Message Processing is complete.

8. Intellectual Property

The IETF 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 this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

9. Acknowledgements

This document is the result of the efforts of the SNMPv3 Working Group. Some special thanks are in order to the following SNMPv3 WG members:

Harald Tveit Alvestrand (Maxware)
Dave Battle (SNMP Research, Inc.)
Alan Beard (Disney Worldwide Services)
Paul Berrevoets (SWI Systemware/Halcyon Inc.)
Martin Bjorklund (Ericsson)
Uri Blumenthal (IBM T. J. Watson Research Center)
Jeff Case (SNMP Research, Inc.)
John Curran (BBN)
Mike Daniele (Compaq Computer Corporation)
T. Max Devlin (Eltrax Systems)
John Flick (Hewlett Packard)
Rob Frye (MCI)
Wes Hardaker (U.C.Davis, Information Technology - D.C.A.S.)
David Harrington (Cabletron Systems Inc.)
Lauren Heintz (BMC Software, Inc.)
N.C. Hien (IBM T. J. Watson Research Center)
Michael Kirkham (InterWorking Labs, Inc.)
Dave Levi (SNMP Research, Inc.)
Louis A Mamakos (UUNET Technologies Inc.)
Joe Marzot (Nortel Networks)
Paul Meyer (Secure Computing Corporation)
Keith McCloghrie (Cisco Systems)
Bob Moore (IBM)
Russ Mundy (TIS Labs at Network Associates)
Bob Natale (ACE*COMM Corporation)
Mike O'Dell (UUNET Technologies Inc.)
Dave Perkins (DeskTalk)
Peter Polkinghorne (Brunel University)
Randy Presuhn (BMC Software, Inc.)
David Reeder (TIS Labs at Network Associates)
David Reid (SNMP Research, Inc.)

Aleksey Romanov (Quality Quorum)
Shawn Routhier (Epilogue)
Juergen Schoenwaelder (TU Braunschweig)
Bob Stewart (Cisco Systems)
Mike Thatcher (Independent Consultant)
Bert Wijnen (IBM T. J. Watson Research Center)

The document is based on recommendations of the IETF Security and Administrative Framework Evolution for SNMP Advisory Team. Members of that Advisory Team were:

David Harrington (Cabletron Systems Inc.)
Jeff Johnson (Cisco Systems)
David Levi (SNMP Research Inc.)
John Linn (Openvision)
Russ Mundy (Trusted Information Systems) chair
Shawn Routhier (Epilogue)
Glenn Waters (Nortel)
Bert Wijnen (IBM T. J. Watson Research Center)

As recommended by the Advisory Team and the SNMPv3 Working Group Charter, the design incorporates as much as practical from previous RFCs and drafts. As a result, special thanks are due to the authors of previous designs known as SNMPv2u and SNMPv2*:

Jeff Case (SNMP Research, Inc.)
David Harrington (Cabletron Systems Inc.)
David Levi (SNMP Research, Inc.)
Keith McCloghrie (Cisco Systems)
Brian O'Keefe (Hewlett Packard)
Marshall T. Rose (Dover Beach Consulting)
Jon Saperia (BGS Systems Inc.)
Steve Waldbusser (International Network Services)
Glenn W. Waters (Bell-Northern Research Ltd.)

10. Security Considerations

The Dispatcher coordinates the processing of messages to provide a level of security for management messages and to direct the SNMP PDUs to the proper SNMP application(s).

A Message Processing Model, and in particular the v3MP defined in this document, interacts as part of the Message Processing with Security Models in the Security Subsystem via the abstract service interface primitives defined in [RFC3411] and elaborated above.

The level of security actually provided is primarily determined by the specific Security Model implementation(s) and the specific SNMP application implementation(s) incorporated into this framework. Applications have access to data which is not secured. Applications should take reasonable steps to protect the data from disclosure, and when they send data across the network, they should obey the securityLevel and call upon the services of an Access Control Model as they apply access control.

The values for the msgID element used in communication between SNMP entities MUST be chosen to avoid replay attacks. The values do not need to be unpredictable; it is sufficient that they not repeat.

When exchanges are carried out over an insecure network, there is an open opportunity for a third party to spoof or replay messages when any message of an exchange is given at the security level of noAuthNoPriv. For most exchanges, all messages exist at the same security level. In the case where the final message is an Internal Class PDU, this message may be delivered at a level of noAuthNoPriv or authNoPriv, independent of the security level of the preceding messages. Internal Class PDUs delivered at the level of authNoPriv are not considered to pose a security hazard. Internal Class PDUs delivered at the security level of noAuthNoPriv open a window of opportunity for spoofing or replay attacks. If the receiver of such messages is aware of these risks, the use of such unauthenticated messages is acceptable and may provide a useful function for discovering engine IDs or for detecting misconfiguration at remote nodes.

This document also contains a MIB definition module. None of the objects defined is writable, and the information they represent is not deemed to be particularly sensitive. However, if they are deemed sensitive in a particular environment, access to them should be restricted through the use of appropriately configured Security and Access Control models.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3411] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3413] Levi, D., Meyer, P. and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.

[RFC3414] Blumenthal, U. and B. Wijnen, "The User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3415] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

[RFC3416] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

[RFC3417] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3417, December 2002.

[RFC3418] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

11.2. Informative References

[RFC1901] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[RFC2028] Hovey, R. and S. Bradner, "The Organizations Involved in the IETF Standards Process", BCP 11, RFC 2028, October 1996.

[RFC2576] Frye, R., Levi, D., Routhier, S. and B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework", RFC 2576, March 2000.

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

12. Editors' Addresses

Jeffrey Case
SNMP Research, Inc.
3001 Kimberlin Heights Road

Knoxville, TN 37920-9716
USA

Phone: +1 423-573-1434
EMail: case@snmp.com

David Harrington
Enterasys Networks
35 Industrial Way
Post Office Box 5005
Rochester, NH 03866-5005
USA

Phone: +1 603-337-2614
EMail: dbh@enterasys.com

Randy Presuhn
BMC Software, Inc.
2141 North First Street
San Jose, CA 95131
USA

Phone: +1 408-546-1006
EMail: randy_presuhn@bmc.com

Bert Wijnen
Lucent Technologies
Schagen 33
3461 GL Linschoten
Netherlands

Phone: +31 348-680-485
EMail: bwijnen@lucent.com

13. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Case, et al.   Standards Track   [Page 43]

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/ref/rfc/v3/rfc3412.txt
No license file was found, but licenses were detected in source scan.

Network Working Group   J. Case
Request for Comments: 1098   University of Tennessee at Knoxville
Obsoletes: RFC 1067   M. Fedor
NYSERNet, Inc.
M. Schoffstall
Rensselaer Polytechnic Institute
C. Davin
MIT Laboratory for Computer Science
April 1989

A Simple Network Management Protocol (SNMP)

Table of Contents

1. Status of this Memo

This RFC is a re-release of RFC 1067, with a changed "Status of this Memo" section. This memo defines a simple protocol by which management information for a network element may be inspected or altered by logically remote users. In particular, together with its companion memos which describe the structure of management information along with the initial management information base, these documents provide a simple, workable architecture and system for managing TCP/IP-based internets and in particular the Internet.

The Internet Activities Board (IAB) has designated two different network management protocols with the same status of "Draft Standard" and "Recommended".

The two protocols are the Common Management Information Services and Protocol over TCP/IP (CMOT) [9], and the Simple Network Management Protocol (SNMP) (this memo).

The IAB intends each of these two protocols to receive the attention of implementers and experimenters. The IAB seeks reports of experience with these two protocols from system builders and users.

By this action, the IAB recommends that all IP and TCP implementations be network manageable (e.g., implement the Internet MIB [3]) and that the implementations that are network manageable are expected to adopt and implement at least one of these two Internet Draft Standards.

Distribution of this memo is unlimited.

2. Introduction

As reported in RFC 1052, IAB Recommendations for the Development of Internet Network Management Standards [1], the Internet Activities Board has directed the Internet Engineering Task Force (IETF) to create two new working groups in the area of network management. One group is charged with the further specification and definition of elements to be included in the Management Information Base (MIB). The other is charged with defining the modifications to the Simple Network Management Protocol (SNMP) to accommodate the short-term

Case, Fedor, Schoffstall, & Davin [Page 2]

RFC 1098 SNMP April 1989

needs of the network vendor and operations communities, and to align with the output of the MIB working group.

The MIB working group has produced two memos, one which defines a Structure for Management Information (SMI) [2] for use by the managed objects contained in the MIB. A second memo [3] defines the list of managed objects.

The output of the SNMP Extensions working group is this memo, which incorporates changes to the initial SNMP definition [4] required to attain alignment with the output of the MIB working group. The changes should be minimal in order to be consistent with the IAB's directive that the working groups be "extremely sensitive to the need to keep the SNMP simple." Although considerable care and debate has gone into the changes to the SNMP which are reflected in this memo, the resulting protocol is not backwardly-compatible with its predecessor, the Simple Gateway Monitoring Protocol (SGMP) [5]. Although the syntax of the protocol has been altered, the original philosophy, design decisions, and architecture remain intact. In order to avoid confusion, new UDP ports have been allocated for use by the protocol described in this memo.

3. The SNMP Architecture

Implicit in the SNMP architectural model is a collection of network management stations and network elements. Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, and the like, which have management agents responsible for performing the network management functions requested by the network management stations. The Simple Network Management Protocol (SNMP) is used to communicate management information between the network management stations and the agents in the network elements.

3.1. Goals of the Architecture

The SNMP explicitly minimizes the number and complexity of management functions realized by the management agent itself. This goal is attractive in at least four respects:

(1) The development cost for management agent software necessary to support the protocol is accordingly reduced.

(2) The degree of management function that is remotely supported is accordingly increased, thereby admitting fullest use of internet resources in the management task.

(3) The degree of management function that is remotely supported is accordingly increased, thereby imposing the fewest possible restrictions on the form and sophistication of management tools.

(4) Simplified sets of management functions are easily understood and used by developers of network management tools.

A second goal of the protocol is that the functional paradigm for monitoring and control be sufficiently extensible to accommodate additional, possibly unanticipated aspects of network operation and management.

A third goal is that the architecture be, as much as possible, independent of the architecture and mechanisms of particular hosts or particular gateways.

3.2. Elements of the Architecture

The SNMP architecture articulates a solution to the network management problem in terms of:

(1) the scope of the management information communicated by the protocol,

(2) the representation of the management information

communicated by the protocol,

(3) operations on management information supported by the protocol,

(4) the form and meaning of exchanges among management entities,

(5) the definition of administrative relationships among management entities, and

(6) the form and meaning of references to management information.

3.2.1. Scope of Management Information

The scope of the management information communicated by operation of the SNMP is exactly that represented by instances of all non-aggregate object types either defined in Internet-standard MIB or defined elsewhere according to the conventions set forth in Internet-standard SMI [2].

Support for aggregate object types in the MIB is neither required for conformance with the SMI nor realized by the SNMP.

3.2.2. Representation of Management Information

Management information communicated by operation of the SNMP is represented according to the subset of the ASN.1 language [6] that is specified for the definition of non-aggregate types in the SMI.

The SGMP adopted the convention of using a well-defined subset of the ASN.1 language [6]. The SNMP continues and extends this tradition by utilizing a moderately more complex subset of ASN.1 for describing managed objects and for describing the protocol data units used for managing those objects. In addition, the desire to ease eventual transition to OSI-based network management protocols led to the definition in the ASN.1 language of an Internet-standard Structure of Management Information (SMI) [2] and Management Information Base (MIB) [3]. The use of the ASN.1 language, was, in part, encouraged by the successful use of ASN.1 in earlier efforts, in particular, the SGMP. The restrictions on the use of ASN.1 that are part of the SMI contribute to the simplicity espoused and validated by experience with the SGMP.

Also for the sake of simplicity, the SNMP uses only a subset of the basic encoding rules of ASN.1 [7]. Namely, all encodings use the definite-length form. Further, whenever permissible, non-constructor encodings are used rather than constructor encodings. This restriction applies to all aspects of ASN.1 encoding, both for the top-level protocol data units and the data objects they contain.

3.2.3. Operations Supported on Management Information

The SNMP models all management agent functions as alterations or inspections of variables. Thus, a protocol entity on a logically remote host (possibly the network element itself) interacts with the management agent resident on the network element in order to retrieve (get) or alter (set) variables. This strategy has at least two positive consequences:

(1) It has the effect of limiting the number of essential management functions realized by the management agent to two: one operation to assign a value to a specified configuration or other parameter and another to retrieve such a value.

(2) A second effect of this decision is to avoid introducing into the protocol definition support for imperative management commands: the number of such commands is in practice ever-increasing, and the semantics of such commands are in general arbitrarily complex.

The strategy implicit in the SNMP is that the monitoring of network state at any significant level of detail is accomplished primarily by polling for appropriate information on the part of the monitoring center(s). A limited number of unsolicited messages (traps) guide the timing and focus of the polling. Limiting the number of unsolicited messages is consistent with the goal of simplicity and minimizing the amount of traffic generated by the network management function.

The exclusion of imperative commands from the set of explicitly supported management functions is unlikely to preclude any desirable management agent operation. Currently, most commands are requests either to set the value of some parameter or to retrieve such a value, and the function of the few imperative commands currently supported is easily accommodated in an asynchronous mode by this management model. In this scheme, an imperative command might be realized as the setting of a parameter value that subsequently triggers the desired action. For example, rather than implementing a "reboot command," this action might be invoked by simply setting a parameter indicating the number of seconds until system reboot.

3.2.4. Form and Meaning of Protocol Exchanges

The communication of management information among management entities is realized in the SNMP through the exchange of protocol messages. The form and meaning of those messages is defined below in Section 4.

Consistent with the goal of minimizing complexity of the management agent, the exchange of SNMP messages requires only an unreliable datagram service, and every message is entirely and independently represented by a single transport datagram. While this document specifies the exchange of messages via the UDP protocol [8], the mechanisms of the SNMP are generally suitable for use with a wide variety of transport services.

3.2.5. Definition of Administrative Relationships

The SNMP architecture admits a variety of administrative relationships among entities that participate in the protocol. The entities residing at management stations and network elements which communicate with one another using the SNMP are termed SNMP application entities. The peer processes which implement the SNMP, and thus support the SNMP application entities, are termed protocol entities.

A pairing of an SNMP agent with some arbitrary set of SNMP application entities is called an SNMP community. Each SNMP community is named by a string of octets, that is called the community name for said community.

An SNMP message originated by an SNMP application entity that in fact belongs to the SNMP community named by the community component of said message is called an authentic SNMP message. The set of rules by which an SNMP message is identified as an authentic SNMP message for a particular SNMP community is called an authentication scheme. An implementation of a function that identifies authentic SNMP messages according to one or more authentication schemes is called an authentication service.

Clearly, effective management of administrative relationships among SNMP application entities requires authentication services that (by the use of encryption or other techniques) are able to identify authentic SNMP messages with a high degree of certainty. Some SNMP implementations may wish to support only a trivial authentication service that identifies all SNMP messages as authentic SNMP messages.

For any network element, a subset of objects in the MIB that pertain to that element is called a SNMP MIB view. Note that the names of the object types represented in a SNMP MIB view need not belong to a

single sub-tree of the object type name space.

An element of the set { READ-ONLY, READ-WRITE } is called an SNMP access mode.

A pairing of a SNMP access mode with a SNMP MIB view is called an SNMP community profile. A SNMP community profile represents specified access privileges to variables in a specified MIB view. For every variable in the MIB view in a given SNMP community profile, access to that variable is represented by the profile according to the following conventions:

(1) if said variable is defined in the MIB with "Access:" of "none," it is unavailable as an operand for any operator;

(2) if said variable is defined in the MIB with "Access:" of "read-write" or "write-only" and the access mode of the given profile is READ-WRITE, that variable is available as an operand for the get, set, and trap operations;

(3) otherwise, the variable is available as an operand for the get and trap operations.

(4) In those cases where a "write-only" variable is an operand used for the get or trap operations, the value given for the variable is implementation-specific.

A pairing of a SNMP community with a SNMP community profile is called a SNMP access policy. An access policy represents a specified community profile afforded by the SNMP agent of a specified SNMP community to other members of that community. All administrative relationships among SNMP application entities are architecturally defined in terms of SNMP access policies.

For every SNMP access policy, if the network element on which the SNMP agent for the specified SNMP community resides is not that to which the MIB view for the specified profile pertains, then that policy is called a SNMP proxy access policy. The SNMP agent associated with a proxy access policy is called a SNMP proxy agent. While careless definition of proxy access policies can result in management loops, prudent definition of proxy policies is useful in at least two ways:

(1) It permits the monitoring and control of network elements which are otherwise not addressable using the management protocol and the transport protocol. That is, a proxy agent may provide a protocol conversion function allowing a management station to apply a consistent management

framework to all network elements, including devices such as modems, multiplexors, and other devices which support different management frameworks.

(2) It potentially shields network elements from elaborate access control policies. For example, a proxy agent may implement sophisticated access control whereby diverse subsets of variables within the MIB are made accessible to different management stations without increasing the complexity of the network element.

By way of example, Figure 1 illustrates the relationship between management stations, proxy agents, and management agents. In this example, the proxy agent is envisioned to be a normal Internet Network Operations Center (INOC) of some administrative domain which has a standard managerial relationship with a set of management agents.

```
+------------------+       +----------------+      +----------------+
|  Region #1 INOC  |       |Region #2 INOC  |      |PC in Region #3 |
|                  |       |                |      |                |
|Domain=Region #1  |       |Domain=Region #2|      |Domain=Region #3|
|CPU=super-mini-1  |       |CPU=super-mini-1|      |CPU=Clone-1     |
|PCommunity=pub    |       |PCommunity=pub  |      |PCommunity=slate|
|                  |       |                |      |                |
+------------------+       +----------------+      +----------------+
       /|\                       /|\                     /|\
        |                         |                       |
        |                         |                       |
        |                        \|/                      |
```

```
     |            +-----------------+            |
     +-------------->| Region #3 INOC  |<-------------+
                  |                 |
                  |Domain=Region #3 |
                  |CPU=super-mini-2 |
                  |PCommunity=pub,  |
                  |        slate    |
                  |DCommunity=secret|
     +-------------->|                 |<-------------+
     |            +-----------------+            |
     |                   /|\                     |
     |                    |                      |
     |                    |                      |
    \|/                  \|/                    \|/
+-----------------+  +-----------------+  +-----------------+
|Domain=Region#3  |  |Domain=Region#3  |  |Domain=Region#3  |
|CPU=router-1     |  |CPU=mainframe-1  |  |CPU=modem-1      |
|DCommunity=secret|  |DCommunity=secret|  |DCommunity=secret|
+-----------------+  +-----------------+  +-----------------+
```

Domain: the administrative domain of the element
PCommunity: the name of a community utilizing a proxy agent
DCommunity: the name of a direct community

Figure 1
Example Network Management Configuration

Case, Fedor, Schoffstall, & Davin [Page 10]

RFC 1098 SNMP April 1989

3.2.6. Form and Meaning of References to Managed Objects

The SMI requires that the definition of a conformant management
protocol address:

(1) the resolution of ambiguous MIB references,

(2) the resolution of MIB references in the presence multiple
MIB versions, and

(3) the identification of particular instances of object
types defined in the MIB.

3.2.6.1. Resolution of Ambiguous MIB References

Because the scope of any SNMP operation is conceptually confined to objects relevant to a single network element, and because all SNMP references to MIB objects are (implicitly or explicitly) by unique variable names, there is no possibility that any SNMP reference to any object type defined in the MIB could resolve to multiple instances of that type.

3.2.6.2. Resolution of References across MIB Versions

The object instance referred to by any SNMP operation is exactly that specified as part of the operation request or (in the case of a get-next operation) its immediate successor in the MIB as a whole. In particular, a reference to an object as part of some version of the Internet-standard MIB does not resolve to any object that is not part of said version of the Internet-standard MIB, except in the case that the requested operation is get-next and the specified object name is lexicographically last among the names of all objects presented as part of said version of the Internet-Standard MIB.

3.2.6.3. Identification of Object Instances

The names for all object types in the MIB are defined explicitly either in the Internet-standard MIB or in other documents which conform to the naming conventions of the SMI. The SMI requires that conformant management protocols define mechanisms for identifying individual instances of those object types for a particular network element.

Each instance of any object type defined in the MIB is identified in SNMP operations by a unique name called its "variable name." In general, the name of an SNMP variable is an OBJECT IDENTIFIER of the form x.y, where x is the name of a non-aggregate object type defined in the MIB and y is an OBJECT IDENTIFIER fragment that, in a way specific to the named object type, identifies the desired instance.

This naming strategy admits the fullest exploitation of the semantics of the GetNextRequest-PDU (see Section 4), because it assigns names for related variables so as to be contiguous in the lexicographical ordering of all variable names known in the MIB.

The type-specific naming of object instances is defined below for a number of classes of object types. Instances of an object type to which none of the following naming conventions are applicable are named by OBJECT IDENTIFIERs of the form x.0, where x is the name of said object type in the MIB definition.

For example, suppose one wanted to identify an instance of the variable sysDescr The object class for sysDescr is:

```
iso org dod internet mgmt mib system sysDescr
 1   3   6     1      2    1     1       1
```

Hence, the object type, x, would be 1.3.6.1.2.1.1.1 to which is appended an instance sub-identifier of 0. That is, 1.3.6.1.2.1.1.1.0 identifies the one and only instance of sysDescr.

3.2.6.3.1. ifTable Object Type Names

The name of a subnet interface, s, is the OBJECT IDENTIFIER value of the form i, where i has the value of that instance of the ifIndex object type associated with s.

For each object type, t, for which the defined name, n, has a prefix of ifEntry, an instance, i, of t is named by an OBJECT IDENTIFIER of the form n.s, where s is the name of the subnet interface about which i represents information.

For example, suppose one wanted to identify the instance of the variable ifType associated with interface 2. Accordingly, ifType.2 would identify the desired instance.

3.2.6.3.2. atTable Object Type Names

The name of an AT-cached network address, x, is an OBJECT IDENTIFIER of the form 1.a.b.c.d, where a.b.c.d is the value (in the familiar "dot" notation) of the atNetAddress object type associated with x.

The name of an address translation equivalence e is an OBJECT IDENTIFIER value of the form s.w, such that s is the value of that instance of the atIndex object type associated with e and such that w is the name of the AT-cached network address associated with e.

For each object type, t, for which the defined name, n, has a prefix of atEntry, an instance, i, of t is named by an OBJECT IDENTIFIER of the form n.y, where y is the name of the address translation equivalence about which i represents information.

For example, suppose one wanted to find the physical address of an entry in the address translation table (ARP cache) associated with an IP address of 89.1.1.42 and interface 3. Accordingly, atPhysAddress.3.1.89.1.1.42 would identify the desired instance.

3.2.6.3.3. ipAddrTable Object Type Names

The name of an IP-addressable network element, x, is the OBJECT IDENTIFIER of the form a.b.c.d such that a.b.c.d is the value (in the familiar "dot" notation) of that instance of the ipAdEntAddr object type associated with x.

For each object type, t, for which the defined name, n, has a prefix of ipAddrEntry, an instance, i, of t is named by an OBJECT IDENTIFIER of the form n.y, where y is the name of the IP-addressable network element about which i represents information.

For example, suppose one wanted to find the network mask of an entry in the IP interface table associated with an IP address of 89.1.1.42. Accordingly, ipAdEntNetMask.89.1.1.42 would identify the desired instance.

3.2.6.3.4. ipRoutingTable Object Type Names

The name of an IP route, x, is the OBJECT IDENTIFIER of the form a.b.c.d such that a.b.c.d is the value (in the familiar "dot" notation) of that instance of the ipRouteDest object type associated with x.

For each object type, t, for which the defined name, n, has a prefix of ipRoutingEntry, an instance, i, of t is named by an OBJECT

IDENTIFIER of the form n.y, where y is the name of the IP route about which i represents information.

For example, suppose one wanted to find the next hop of an entry in the IP routing table associated with the destination of 89.1.1.42. Accordingly, ipRouteNextHop.89.1.1.42 would identify the desired instance.

3.2.6.3.5. tcpConnTable Object Type Names

The name of a TCP connection, x, is the OBJECT IDENTIFIER of the form a.b.c.d.e.f.g.h.i.j such that a.b.c.d is the value (in the familiar

Case, Fedor, Schoffstall, & Davin [Page 13]

RFC 1098 SNMP April 1989

"dot" notation) of that instance of the tcpConnLocalAddress object type associated with x and such that f.g.h.i is the value (in the familiar "dot" notation) of that instance of the tcpConnRemoteAddress object type associated with x and such that e is the value of that instance of the tcpConnLocalPort object type associated with x and such that j is the value of that instance of the tcpConnRemotePort object type associated with x.

For each object type, t, for which the defined name, n, has a prefix of tcpConnEntry, an instance, i, of t is named by an OBJECT IDENTIFIER of the form n.y, where y is the name of the TCP connection about which i represents information.

For example, suppose one wanted to find the state of a TCP connection between the local address of 89.1.1.42 on TCP port 21 and the remote address of 10.0.0.51 on TCP port 2059. Accordingly, tcpConnState.89.1.1.42.21.10.0.0.51.2059 would identify the desired instance.

3.2.6.3.6. egpNeighTable Object Type Names

The name of an EGP neighbor, x, is the OBJECT IDENTIFIER of the form a.b.c.d such that a.b.c.d is the value (in the familiar "dot" notation) of that instance of the egpNeighAddr object type associated with x.

For each object type, t, for which the defined name, n, has a prefix of egpNeighEntry, an instance, i, of t is named by an OBJECT

IDENTIFIER of the form n.y, where y is the name of the EGP neighbor about which i represents information.

For example, suppose one wanted to find the neighbor state for the IP address of 89.1.1.42. Accordingly, egpNeighState.89.1.1.42 would identify the desired instance.

4. Protocol Specification

The network management protocol is an application protocol by which the variables of an agent's MIB may be inspected or altered.

Communication among protocol entities is accomplished by the exchange of messages, each of which is entirely and independently represented within a single UDP datagram using the basic encoding rules of ASN.1 (as discussed in Section 3.2.2). A message consists of a version identifier, an SNMP community name, and a protocol data unit (PDU). A protocol entity receives messages at UDP port 161 on the host with which it is associated for all messages except for those which report traps (i.e., all messages except those which contain the Trap-PDU). Messages which report traps should be received on UDP port 162 for further processing. An implementation of this protocol need not accept messages whose length exceeds 484 octets. However, it is recommended that implementations support larger datagrams whenever feasible.

It is mandatory that all implementations of the SNMP support the five

PDUs: GetRequest-PDU, GetNextRequest-PDU, GetResponse-PDU,
SetRequest-PDU, and Trap-PDU.

```
RFC1098-SNMP DEFINITIONS ::= BEGIN

 IMPORTS
     ObjectName, ObjectSyntax, NetworkAddress, IpAddress, TimeTicks
          FROM RFC1065-SMI;


 -- top-level message

      Message ::=
           SEQUENCE {
               version        -- version-1 for this RFC
                  INTEGER {
                     version-1(0)
                  },

               community      -- community name
                  OCTET STRING,

               data           -- e.g., PDUs if trivial
                  ANY         -- authentication is being used
           }
```



```
 -- protocol data units

      PDUs ::=
           CHOICE {
               get-request
                  GetRequest-PDU,

               get-next-request
                  GetNextRequest-PDU,

               get-response
                  GetResponse-PDU,
```

```
set-request
   SetRequest-PDU,

trap
   Trap-PDU
 }

-- the individual PDUs and commonly used
-- data types will be defined later

END
```

4.1. Elements of Procedure

This section describes the actions of a protocol entity implementing the SNMP. Note, however, that it is not intended to constrain the internal architecture of any conformant implementation.

In the text that follows, the term transport address is used. In the case of the UDP, a transport address consists of an IP address along with a UDP port. Other transport services may be used to support the SNMP. In these cases, the definition of a transport address should be made accordingly.

The top-level actions of a protocol entity which generates a message are as follows:

(1) It first constructs the appropriate PDU, e.g., the GetRequest-PDU, as an ASN.1 object.

(2) It then passes this ASN.1 object along with a community name its source transport address and the destination transport address, to the service which implements the desired authentication scheme. This authentication

service returns another ASN.1 object.

(3) The protocol entity then constructs an ASN.1 Message object, using the community name and the resulting ASN.1

object.

(4) This new ASN.1 object is then serialized, using the basic encoding rules of ASN.1, and then sent using a transport service to the peer protocol entity.

Similarly, the top-level actions of a protocol entity which receives a message are as follows:

(1) It performs a rudimentary parse of the incoming datagram to build an ASN.1 object corresponding to an ASN.1 Message object. If the parse fails, it discards the datagram and performs no further actions.

(2) It then verifies the version number of the SNMP message. If there is a mismatch, it discards the datagram and performs no further actions.

(3) The protocol entity then passes the community name and user data found in the ASN.1 Message object, along with the datagram's source and destination transport addresses to the service which implements the desired authentication scheme. This entity returns another ASN.1 object, or signals an authentication failure. In the latter case, the protocol entity notes this failure, (possibly) generates a trap, and discards the datagram and performs no further actions.

(4) The protocol entity then performs a rudimentary parse on the ASN.1 object returned from the authentication service to build an ASN.1 object corresponding to an ASN.1 PDUs object. If the parse fails, it discards the datagram and performs no further actions. Otherwise, using the named SNMP community, the appropriate profile is selected, and the PDU is processed accordingly. If, as a result of this processing, a message is returned then the source transport address that the response message is sent from shall be identical to the destination transport address that the original request message was sent to.

4.1.1. Common Constructs

Before introducing the six PDU types of the protocol, it is appropriate to consider some of the ASN.1 constructs used frequently:

```
-- request/response information

RequestID ::=
        INTEGER

ErrorStatus ::=
        INTEGER {
            noError(0),
            tooBig(1),
            noSuchName(2),
            badValue(3),
            readOnly(4)
            genErr(5)
        }

ErrorIndex ::=
        INTEGER


-- variable bindings

VarBind ::=
        SEQUENCE {
            name
                ObjectName,

            value
                ObjectSyntax
        }

VarBindList ::=
        SEQUENCE OF
            VarBind
```

RequestIDs are used to distinguish among outstanding requests. By use of the RequestID, an SNMP application entity can correlate incoming responses with outstanding requests. In cases where an unreliable datagram service is being used, the RequestID also provides a simple means of identifying messages duplicated by the network.

A non-zero instance of ErrorStatus is used to indicate that an

exception occurred while processing a request. In these cases, ErrorIndex may provide additional information by indicating which variable in a list caused the exception.

The term variable refers to an instance of a managed object. A variable binding, or VarBind, refers to the pairing of the name of a variable to the variable's value. A VarBindList is a simple list of variable names and corresponding values. Some PDUs are concerned only with the name of a variable and not its value (e.g., the GetRequest-PDU). In this case, the value portion of the binding is ignored by the protocol entity. However, the value portion must still have valid ASN.1 syntax and encoding. It is recommended that the ASN.1 value NULL be used for the value portion of such bindings.

4.1.2. The GetRequest-PDU

The form of the GetRequest-PDU is:

```
GetRequest-PDU ::=
    [0]
      IMPLICIT SEQUENCE {
          request-id
              RequestID,

          error-status        -- always 0
              ErrorStatus,

          error-index         -- always 0
              ErrorIndex,

          variable-bindings
              VarBindList
      }
```

The GetRequest-PDU is generated by a protocol entity only at the request of its SNMP application entity.

Upon receipt of the GetRequest-PDU, the receiving protocol entity responds according to any applicable rule in the list below:

(1) If, for any object named in the variable-bindings field, the object's name does not exactly match the name of some object available for get operations in the relevant MIB view, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is noSuchName, and the value of the error-index field is the index of said object name component in the received

message.

(2) If, for any object named in the variable-bindings field, the object is an aggregate type (as defined in the SMI), then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is noSuchName, and the value of the error-index field is the index of said object name component in the received message.

(3) If the size of the GetResponse-PDU generated as described below would exceed a local limitation, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is tooBig, and the value of the error-index field is zero.

(4) If, for any object named in the variable-bindings field, the value of the object cannot be retrieved for reasons not covered by any of the foregoing rules, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is genErr and the value of the error-index field is the index of said object name component in the received message.

If none of the foregoing rules apply, then the receiving protocol entity sends to the originator of the received message the GetResponse-PDU such that, for each object named in the variable-bindings field of the received message, the corresponding component of the GetResponse-PDU represents the name and value of that variable. The value of the error- status field of the GetResponse-PDU is noError and the value of the error-index field is zero. The value of the request-id field of the GetResponse-PDU is that of the received message.

4.1.3. The GetNextRequest-PDU

The form of the GetNextRequest-PDU is identical to that of the GetRequest-PDU except for the indication of the PDU type. In the ASN.1 language:

```
GetNextRequest-PDU ::=
    [1]
        IMPLICIT SEQUENCE {
            request-id
                RequestID,
```



```
            error-status        -- always 0
                ErrorStatus,

            error-index         -- always 0
                ErrorIndex,

            variable-bindings
                VarBindList
        }
```

The GetNextRequest-PDU is generated by a protocol entity only at the request of its SNMP application entity.

Upon receipt of the GetNextRequest-PDU, the receiving protocol entity responds according to any applicable rule in the list below:

(1) If, for any object name in the variable-bindings field, that name does not lexicographically precede the name of some object available for get operations in the relevant

MIB view, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is noSuchName, and the value of the error-index field is the index of said object name component in the received message.

(2) If the size of the GetResponse-PDU generated as described below would exceed a local limitation, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is tooBig, and the value of the error-index field is zero.

(3) If, for any object named in the variable-bindings field, the value of the lexicographical successor to the named object cannot be retrieved for reasons not covered by any of the foregoing rules, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is genErr and the value of the error-index field is the index of said object name component in the received message.

If none of the foregoing rules apply, then the receiving protocol entity sends to the originator of the received message the GetResponse-PDU such that, for each name in the variable-bindings field of the received message, the corresponding component of the

GetResponse-PDU represents the name and value of that object whose name is, in the lexicographical ordering of the names of all objects available for get operations in the relevant MIB view, together with the value of the name field of the given component, the immediate successor to that value. The value of the error-status field of the GetResponse-PDU is noError and the value of the errorindex field is zero. The value of the request-id field of the GetResponse-PDU is that of the received message.

4.1.3.1. Example of Table Traversal

One important use of the GetNextRequest-PDU is the traversal of conceptual tables of information within the MIB. The semantics of this type of SNMP message, together with the protocol-specific mechanisms for identifying individual instances of object types in the MIB, affords access to related objects in the MIB as if they enjoyed a tabular organization.

By the SNMP exchange sketched below, an SNMP application entity might extract the destination address and next hop gateway for each entry in the routing table of a particular network element. Suppose that this routing table has three entries:

Destination	NextHop	Metric
10.0.0.99	89.1.1.42	5
9.1.2.3	99.0.0.3	3
10.0.0.51	89.1.1.42	5

The management station sends to the SNMP agent a GetNextRequest-PDU containing the indicated OBJECT IDENTIFIER values as the requested variable names:

GetNextRequest (ipRouteDest, ipRouteNextHop, ipRouteMetric1)

The SNMP agent responds with a GetResponse-PDU:

```
GetResponse (( ipRouteDest.9.1.2.3 =  "9.1.2.3" ),
      ( ipRouteNextHop.9.1.2.3 = "99.0.0.3" ),
      ( ipRouteMetric1.9.1.2.3 = 3 ))
```

The management station continues with:

```
GetNextRequest ( ipRouteDest.9.1.2.3,
      ipRouteNextHop.9.1.2.3,
```



```
      ipRouteMetric1.9.1.2.3 )
```

The SNMP agent responds:

```
GetResponse (( ipRouteDest.10.0.0.51 = "10.0.0.51" ),
       ( ipRouteNextHop.10.0.0.51 = "89.1.1.42" ),
       ( ipRouteMetric1.10.0.0.51 = 5 ))
```

The management station continues with:

```
GetNextRequest ( ipRouteDest.10.0.0.51,
       ipRouteNextHop.10.0.0.51,
       ipRouteMetric1.10.0.0.51 )
```

The SNMP agent responds:

```
GetResponse (( ipRouteDest.10.0.0.99 = "10.0.0.99" ),
       ( ipRouteNextHop.10.0.0.99 = "89.1.1.42" ),
       ( ipRouteMetric1.10.0.0.99 = 5 ))
```

The management station continues with:

```
GetNextRequest ( ipRouteDest.10.0.0.99,
       ipRouteNextHop.10.0.0.99,
       ipRouteMetric1.10.0.0.99 )
```

As there are no further entries in the table, the SNMP agent returns those objects that are next in the lexicographical ordering of the known object names. This response signals the end of the routing table to the management station.

4.1.4. The GetResponse-PDU

The form of the GetResponse-PDU is identical to that of the GetRequest-PDU except for the indication of the PDU type. In the ASN.1 language:

```
GetResponse-PDU ::=
    [2]
        IMPLICIT SEQUENCE {
            request-id
                RequestID,
```

```
            error-status
               ErrorStatus,

            error-index
               ErrorIndex,

            variable-bindings
               VarBindList
         }
```

The GetResponse-PDU is generated by a protocol entity only upon receipt of the GetRequest-PDU, GetNextRequest-PDU, or SetRequest-PDU, as described elsewhere in this document.

Upon receipt of the GetResponse-PDU, the receiving protocol entity presents its contents to its SNMP application entity.

4.1.5. The SetRequest-PDU

The form of the SetRequest-PDU is identical to that of the GetRequest-PDU except for the indication of the PDU type. In the ASN.1 language:

```
     SetRequest-PDU ::=
        [3]
         IMPLICIT SEQUENCE {
            request-id
               RequestID,

            error-status       -- always 0
               ErrorStatus,

            error-index        -- always 0
               ErrorIndex,

            variable-bindings
               VarBindList
         }
```

The SetRequest-PDU is generated by a protocol entity only at the request of its SNMP application entity.

Upon receipt of the SetRequest-PDU, the receiving entity responds according to any applicable rule in the list below:

(1) If, for any object named in the variable-bindings field,

the object is not available for set operations in the relevant MIB view, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is noSuchName, and the value of the error-index field is the index of said object name component in the received message.

(2) If, for any object named in the variable-bindings field, the contents of the value field does not, according to the ASN.1 language, manifest a type, length, and value that is consistent with that required for the variable, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is badValue, and the value of the error-index field is the index of said object name in the received message.

(3) If the size of the Get Response type message generated as described below would exceed a local limitation, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is tooBig, and the value of the error-index field is zero.

(4) If, for any object named in the variable-bindings field, the value of the named object cannot be altered for reasons not covered by any of the foregoing rules, then the receiving entity sends to the originator of the received message the GetResponse-PDU of identical form, except that the value of the error-status field is genErr and the value of the error-index field is the index of said object name component in the received message.

If none of the foregoing rules apply, then for each object named in the variable-bindings field of the received message, the corresponding value is assigned to the variable. Each variable assignment specified by the SetRequest-PDU should be effected as if simultaneously set with respect to all other assignments specified in the same message.

The receiving entity then sends to the originator of the received message the GetResponse-PDU of identical form except that the value of the error-status field of the generated message is noError and the value of the error-index field is zero.

4.1.6. The Trap-PDU

The form of the Trap-PDU is:

```
Trap-PDU ::=
    [4]

      IMPLICIT SEQUENCE {
        enterprise          -- type of object generating
                            -- trap, see sysObjectID in [2]
          OBJECT IDENTIFIER,

        agent-addr          -- address of object generating
          NetworkAddress,   -- trap

        generic-trap        -- generic trap type
          INTEGER {
            coldStart(0),
            warmStart(1),
            linkDown(2),
            linkUp(3),
            authenticationFailure(4),
            egpNeighborLoss(5),
            enterpriseSpecific(6)
          },

        specific-trap     -- specific code, present even
          INTEGER,        -- if generic-trap is not
```

```
                  -- enterpriseSpecific

     time-stamp      -- time elapsed between the last
       TimeTicks,     -- (re)initialization of the network
                  -- entity and the generation of the
                     trap

     variable-bindings   -- "interesting" information
          VarBindList
  }
```

The Trap-PDU is generated by a protocol entity only at the request of the SNMP application entity. The means by which an SNMP application entity selects the destination addresses of the SNMP application entities is implementation-specific.

Upon receipt of the Trap-PDU, the receiving protocol entity presents its contents to its SNMP application entity.

The significance of the variable-bindings component of the Trap-PDU is implementation-specific.

Interpretations of the value of the generic-trap field are:

4.1.6.1. The coldStart Trap

A coldStart(0) trap signifies that the sending protocol entity is reinitializing itself such that the agent's configuration or the protocol entity implementation may be altered.

4.1.6.2. The warmStart Trap

A warmStart(1) trap signifies that the sending protocol entity is reinitializing itself such that neither the agent configuration nor the protocol entity implementation is altered.

4.1.6.3. The linkDown Trap

A linkDown(2) trap signifies that the sending protocol entity recognizes a failure in one of the communication links represented in the agent's configuration.

The Trap-PDU of type linkDown contains as the first element of its variable-bindings, the name and value of the ifIndex instance for the affected interface.

4.1.6.4. The linkUp Trap

A linkUp(3) trap signifies that the sending protocol entity recognizes that one of the communication links represented in the agent's configuration has come up.

The Trap-PDU of type linkUp contains as the first element of its variable-bindings, the name and value of the ifIndex instance for the affected interface.

4.1.6.5. The authenticationFailure Trap

An authenticationFailure(4) trap signifies that the sending protocol entity is the addressee of a protocol message that is not properly authenticated. While implementations of the SNMP must be capable of generating this trap, they must also be capable of suppressing the emission of such traps via an implementation-specific mechanism.

4.1.6.6. The egpNeighborLoss Trap

An egpNeighborLoss(5) trap signifies that an EGP neighbor for whom

Case, Fedor, Schoffstall, & Davin [Page 27]

RFC 1098 SNMP April 1989

the sending protocol entity was an EGP peer has been marked down and the peer relationship no longer obtains.

The Trap-PDU of type egpNeighborLoss contains as the first element of its variable-bindings, the name and value of the egpNeighAddr instance for the affected neighbor.

4.1.6.7. The enterpriseSpecific Trap

A enterpriseSpecific(6) trap signifies that the sending protocol entity recognizes that some enterprise-specific event has occurred. The specific-trap field identifies the particular trap which occurred.

5. Definitions

RFC1098-SNMP DEFINITIONS ::= BEGIN

```
IMPORTS
    ObjectName, ObjectSyntax, NetworkAddress, IpAddress, TimeTicks
        FROM RFC1065-SMI;


    -- top-level message

    Message ::=
            SEQUENCE {
                version          -- version-1 for this RFC
                    INTEGER {
                        version-1(0)
                    },

                community        -- community name
                    OCTET STRING,

                data             -- e.g., PDUs if trivial
                    ANY          -- authentication is being used
            }


    -- protocol data units

    PDUs ::=
            CHOICE {
                        get-request
                            GetRequest-PDU,

                        get-next-request
                            GetNextRequest-PDU,

                        get-response
                            GetResponse-PDU,

                        set-request
                            SetRequest-PDU,

                        trap
                            Trap-PDU
                    }
```

```
-- PDUs

GetRequest-PDU ::=
   [0]
      IMPLICIT PDU

GetNextRequest-PDU ::=
   [1]
      IMPLICIT PDU

GetResponse-PDU ::=
   [2]
      IMPLICIT PDU

SetRequest-PDU ::=
   [3]
      IMPLICIT PDU

PDU ::=
      SEQUENCE {
        request-id
            INTEGER,

         error-status      -- sometimes ignored
            INTEGER {
               noError(0),
               tooBig(1),
               noSuchName(2),
               badValue(3),
               readOnly(4),
               genErr(5)
            },

         error-index       -- sometimes ignored
           INTEGER,

         variable-bindings -- values are sometimes ignored
            VarBindList
      }

Trap-PDU ::=
   [4]
     IMPLICIT SEQUENCE {
         enterprise        -- type of object generating
```

```
                     -- trap, see sysObjectID in [2]

                OBJECT IDENTIFIER,
```



```
               agent-addr      -- address of object generating
                  NetworkAddress, -- trap

               generic-trap     -- generic trap type
                  INTEGER {
                     coldStart(0),
                     warmStart(1),
                     linkDown(2),
                     linkUp(3),
                     authenticationFailure(4),
                     egpNeighborLoss(5),
                     enterpriseSpecific(6)
                  },

               specific-trap  -- specific code, present even
                  INTEGER,   -- if generic-trap is not
                             -- enterpriseSpecific

               time-stamp     -- time elapsed between the last
                  TimeTicks, -- (re)initialization of the
                               network
                             -- entity and the generation of the
                               trap

                variable-bindings -- "interesting" information
                  VarBindList
             }


        -- variable bindings

        VarBind ::=
             SEQUENCE {
               name
                  ObjectName,
```

```
        value
            ObjectSyntax
    }

VarBindList ::=
        SEQUENCE OF
           VarBind

END
```


6. Acknowledgements

This memo was influenced by the IETF SNMP Extensions working group:

Karl Auerbach, Epilogue Technology
K. Ramesh Babu, Excelan
Amatzia Ben-Artzi, 3Com/Bridge
Lawrence Besaw, Hewlett-Packard
Jeffrey D. Case, University of Tennessee at Knoxville
Anthony Chung, Sytek
James Davidson, The Wollongong Group
James R. Davin, MIT Laboratory for Computer Science
Mark S. Fedor, NYSERNet
Phill Gross, The MITRE Corporation
Satish Joshi, ACC
Dan Lynch, Advanced Computing Environments
Keith McCloghrie, The Wollongong Group
Marshall T. Rose, The Wollongong Group (chair)
Greg Satz, cisco
Martin Lee Schoffstall, Rensselaer Polytechnic Institute
Wengyik Yeong, NYSERNet

7. References

[1] Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards", RFC 1052, IAB, April 1988.

[2] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", RFC 1065, TWG, August 1988.

[3] McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1066, TWG, August 1988.

[4] Case, J., M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol", Internet Engineering Task Force working note, Network Information Center, SRI International, Menlo Park, California, March 1988.

[5] Davin, J., J. Case, M. Fedor, and M. Schoffstall, "A Simple Gateway Monitoring Protocol", RFC 1028, Proteon, University of Tennessee at Knoxville, Cornell University, and Rensselaer Polytechnic Institute, November 1987.

[6] Information processing systems - Open Systems Interconnection, "Specification of Abstract Syntax Notation One (ASN.1)", International Organization for Standardization, International Standard 8824, December 1987.

[7] Information processing systems - Open Systems Interconnection, "Specification of Basic Encoding Rules for Abstract Notation One (ASN.1)", International Organization for Standardization, International Standard 8825, December 1987.

[8] Postel, J., "User Datagram Protocol", RFC 768, USC/Information Sciences Institute, November 1980.

[9] Warrier, U., and L. Besaw, "The Common Management Information Services and Protocol over TCP/IP", RFC 1095, Unisys Corporation and Hewlett-Packard, April 1989.

Authors' Addresses

Jeffrey D. Case
University of Tennessee Computing Center
Associate Driector
200 Stokely Management Center
Knoxville, TN 37996-0520

Phone: (615) 974-6721

Email: case@UTKUX1.UTK.EDU

Mark Fedor
Nysernet, Inc.
Rensselaer Technology Park
125 Jordan Road
Troy, NY 12180

Phone: (518) 283-8860

Email: fedor@patton.NYSER.NET

Martin Lee Schoffstall
NYSERNET Inc.
Rensselaer Technology Park
165 Jordan Road
Troy, NY 12180

Phone: (518) 283-8860

Email: schoff@NISC.NYSER.NET

Chuck Davin
MIT Laboratory for Computer Science, NE43-507
545 Technology Square
Cambridge, MA 02139

Phone: (617) 253-6020

EMail: jrd@ptt.lcs.mit.edu

Found in path(s):
* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/ref/rfc/v1/rfc1098.txt
No license file was found, but licenses were detected in source scan.

Network Working Group K. McCloghrie
Request For Comments: 1066 M. Rose
TWG

August 1988

Management Information Base for Network Management
of TCP/IP-based internets

Table of Contents

1. Status of this Memo

This memo provides the initial version of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets in the short-term. In particular, together with its companion memos which describe the structure of management information along with the initial network management protocol, these documents provide a simple, workable architecture and system for managing TCP/IP-based internets and in particular the Internet.

This memo specifies a draft standard for the Internet community. TCP/IP implementations in the Internet which are network manageable are expected to adopt and implement this specification.

Distribution of this memo is unlimited.

2. IAB POLICY STATEMENT

This MIB specification is the first edition of an evolving document defining variables needed for monitoring and control of various components of the Internet. Not all groups of defined variables are mandatory for all Internet components.

For example, the EGP group is mandatory for gateways using EGP but not for hosts which should not be running EGP. Similarly, the TCP group is mandatory for hosts running TCP but not for gateways which aren't running it. What IS mandatory, however, is that all variables of a group be supported if any element of the group is supported.

It is expected that additional MIB groups and variables will be defined over time to accommodate the monitoring and control needs of new or changing components of the Internet. The MIB working group will continue to refine this specification and projects a revision incorporating new requirements in early 1989.

3. Introduction

As reported in RFC 1052, IAB Recommendations for the Development of Internet Network Management Standards [1], the Internet Activities Board has directed the Internet Engineering Task Force (IETF) to create two new working groups in the area of network management. One group is charged with the further specification and definition of elements to be included in the Management Information Base. The other is charged with defining the modifications to the Simple Network Management Protocol (SNMP) to accommodate the short-term needs of the network vendor and operator communities. The long-term needs of the Internet community are to be met using the ISO CMIS/CMIP [2,3] framework as a basis. An existing IETF working group, the "NETMAN" group, is already engaged in defining the use of CMIS/CMIP in a TCP/IP network, and will continue with responsibility for addressing the longer-term requirements.

The output of the MIB working group is to be provided to both the

SNMP working group and the NETMAN group, so as to ensure compatibility of monitored items for both network management frameworks.

The MIB working group has produced this memo and a companion. The

companion memo [4] defines a Structure for Management Information (SMI) for use by the managed objects contained in the MIB. This memo defines the list of managed objects.

The IAB also urged the working groups to be "extremely sensitive to the need to keep SNMP simple," and recommends that the MIB working group take as its starting inputs the MIB definitions found in the High-Level Entity Management Systems (HEMS) RFC 1024 [5], the initial SNMP specification [6], and the CMIS/CMIP memos [7,8].

Thus, the list of managed objects defined here, has been derived by taking only those elements which are considered essential. Since such elements are essential, there is no need to allow the implementation of individual objects, to be optional. Rather, all compliant implementations will contain all applicable (see below) objects defined in this memo.

This approach of taking only the essential objects is NOT restrictive, since the SMI defined in the companion memo provides three extensibility mechanisms: one, the addition of new standard objects through the definitions of new versions of the MIB; two, the addition of widely-available but non-standard objects through the multilateral subtree; and three, the addition of private objects through the enterprises subtree. Such additional objects can not only be used for vendor-specific elements, but also for experimentation as required to further the knowledge of which other objects are essential.

The primary criterion for being considered essential was for an object to be contained in all of the above referenced MIB definitions. A few other objects have been included, but only if the MIB working group believed they are truly essential. The detailed list of criteria against which potential inclusions in this (initial) MIB were considered, was:

1) An object needed to be essential for either fault or configuration management.

2) Only weak control objects were permitted (by weak, it is meant that tampering with them can do only limited damage). This criterion reflects the fact that the current management protocols are not sufficiently secure to do more powerful control operations.

3) Evidence of current use and utility was required.

4) An attempt was made to limit the number of objects to about 100 to make it easier for vendors to fully

instrument their software.

5) To avoid redundant variables, it was required that no object be included that can be derived from others in the MIB.

6) Implementation specific objects (e.g., for BSD UNIX) were excluded.

7) It was agreed to avoid heavily instrumenting critical sections of code. The general guideline was one counter per critical section per layer.

4. Objects

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using Abstract Syntax Notation One (ASN.1) [9].

The mechanisms used for describing these objects are specified in the companion memo. In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the OBJECT DESCRIPTOR, to also refer to the object type.

The syntax of an object type defines the abstract data structure corresponding to that object type. The ASN.1 language is used for this purpose. However, the companion memo purposely restricts the ASN.1 constructs which may be used. These restrictions are explicitly made for simplicity.

The encoding of an object type is simply how that object type is represented using the object type's syntax. Implicitly tied to the notion of an object type's syntax and encoding is how the object type is represented when being transmitted on the network. This memo specifies the use of the basic encoding rules of ASN.1 [10].

4.1. Object Groups

Since this list of managed objects contains only the essential elements, there is no need to allow individual objects to be optional. Rather, the objects are arranged into the following groups:

- System
- Interfaces
- Address Translation
- IP
- ICMP
- TCP
- UDP
- EGP

There are two reasons for defining these groups: one, to provide a means of assigning object identifiers; two, to provide a method for implementations of managed agents to know which objects they must implement. This method is as follows: if the semantics of a group is applicable to an implementation, then it must implement all objects

in that group. For example, an implementation must implement the EGP group if and only if it implements the EGP protocol.

4.2. Format of Definitions

The next section contains the specification of all object types contained in the MIB. Following the conventions of the companion memo, the object types are defined using the following fields:

OBJECT:

A textual name, termed the OBJECT DESCRIPTOR, for the object type, along with its corresponding OBJECT IDENTIFIER.

Syntax:

The abstract syntax for the object type, presented using ASN.1. This must resolve to an instance of the ASN.1 type ObjectSyntax defined in the SMI.

Definition:

A textual description of the semantics of the object type. Implementations should ensure that their interpretation of the object type fulfills this definition since this MIB is intended for use in multi-vendor environments. As such it is vital that object types have consistent meaning across all machines.

Access:

One of read-only, read-write, write-only, or not-accessible.

Status:

One of mandatory, optional, or obsolete.

5. Object Definitions

RFC1066-MIB { iso org(3) dod(6) internet(1) mgmt(2) 1 }

```
DEFINITIONS ::= BEGIN

IMPORTS
        mgmt, OBJECT-TYPE, NetworkAddress, IpAddress,
        Counter, Gauge, TimeTicks
            FROM RFC1065-SMI;

mib        OBJECT IDENTIFIER ::= { mgmt 1 }

system     OBJECT IDENTIFIER ::= { mib 1 }
interfaces OBJECT IDENTIFIER ::= { mib 2 }
at         OBJECT IDENTIFIER ::= { mib 3 }
ip         OBJECT IDENTIFIER ::= { mib 4 }
icmp       OBJECT IDENTIFIER ::= { mib 5 }
tcp        OBJECT IDENTIFIER ::= { mib 6 }
udp        OBJECT IDENTIFIER ::= { mib 7 }
egp        OBJECT IDENTIFIER ::= { mib 8 }

END
```

5.1. The System Group

Implementation of the System group is mandatory for all systems.

OBJECT:
----
sysDescr { system 1 }

Syntax:
OCTET STRING

Definition:
A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
sysObjectID { system 2 }

Syntax:
OBJECT IDENTIFIER

Definition:
The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining "what kind of box" is being managed. For example, if vendor "Flintstones, Inc." was assigned the subtree 1.3.6.1.4.1.42, it could assign the identifier 1.3.6.1.4.1.42.1.1 to its "Fred Router".

Access:
read-only.

Status:
mandatory.

OBJECT:
----
sysUpTime { system 3 }

Syntax:
TimeTicks

Definition:
The time (in hundredths of a second) since the network management portion of the system was last re-initialized.

Access:
read-only.

Status:
mandatory.

McCloghrie & Rose [Page 9]

RFC 1066 MIB August 1988

5.2. The Interfaces Group

Implementation of the Interfaces group is mandatory for all systems.

OBJECT:
----
ifNumber { interfaces 1 }

Syntax:
INTEGER

Definition:
The number of network interfaces (regardless of their current state) on which this system can send/receive IP datagrams.

Access:
read-only.

Status:
mandatory.

5.2.1. The Interfaces Table

OBJECT:
----

ifTable { interfaces 2 }

Syntax:
SEQUENCE OF IfEntry

Definition:
A list of interface entries. The number of entries is given by the value of ifNumber.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
ifEntry { ifTable 1 }

Syntax:
IfEntry ::= SEQUENCE {


```
ifIndex
  INTEGER,
ifDescr
  OCTET STRING,
ifType
  INTEGER,
ifMtu
  INTEGER,
ifSpeed
  Gauge,
ifPhysAddress
  OCTET STRING,
ifAdminStatus
  INTEGER,
ifOperStatus
  INTEGER,
ifLastChange
  TimeTicks,
ifInOctets
```

```
        Counter,
    ifInUcastPkts
        Counter,
    ifInNUcastPkts
        Counter,
    ifInDiscards
        Counter,
    ifInErrors
        Counter,
    ifInUnknownProtos
        Counter,
    ifOutOctets
        Counter,
    ifOutUcastPkts
        Counter,
    ifOutNUcastPkts
        Counter,
    ifOutDiscards
        Counter,
    ifOutErrors
        Counter,
    ifOutQLen
        Gauge
}
```

Definition:
An interface entry containing objects at the subnetwork layer and below for a particular interface.

Access:
read-write.

Status:
mandatory.

We now consider the individual components of each interface entry:

OBJECT:
----
ifIndex { ifEntry 1 }

Syntax:
INTEGER

Definition:
A unique value for each interface. Its value ranges between 1 and the value of ifNumber. The value for each interface must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifDescr { ifEntry 2 }

Syntax:
OCTET STRING

Definition:
A text string containing information about the interface. This string should include the name of the manufacturer, the product name and the version of the hardware interface. The string is intended for presentation to a human; it must not contain anything but printable ASCII characters.

McCloghrie & Rose [Page 12]

RFC 1066 MIB August 1988

Access:
read-only.

Status:
    mandatory.

OBJECT:
----
    ifType { ifEntry 3 }

Syntax:
```
INTEGER {
    other(1),          -- none of the following
    regular1822(2),
    hdh1822(3),
    ddn-x25(4),
    rfc877-x25(5),
    ethernet-csmacd(6),
    iso88023-csmacd(7),
    iso88024-tokenBus(8),
    iso88025-tokenRing(9),
    iso88026-man(10),
    starLan(11),
    proteon-10MBit(12),
    proteon-80MBit(13),
    hyperchannel(14),
    fddi(15),
    lapb(16),
    sdlc(17),
    t1-carrier(18),
    cept(19),          -- european equivalent of T-1
    basicIsdn(20),
    primaryIsdn(21),
                       -- proprietary serial
    propPointToPointSerial(22)
}
```

Definition:
    The type of interface, distinguished according to the
    physical/link/network protocol(s) immediately "below" IP
    in the protocol stack.

Access:
    read-only.

Status:
    mandatory.

OBJECT:
----
ifMtu { ifEntry 4 }

Syntax:
INTEGER

Definition:
The size of the largest IP datagram which can be sent/received on the interface, specified in octets.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifSpeed { ifEntry 5 }

Syntax:
Gauge

Definition:
An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifPhysAddress { ifEntry 6 }

Syntax:
    OCTET STRING

Definition:
    The interface's address at the protocol layer immediately

    "below" IP in the protocol stack. For interfaces which
    do not have such an address (e.g., a serial line), this
    object should contain an octet string of zero length.

Access:
    read-only.

Status:
    mandatory.

OBJECT:
----
    ifAdminStatus { ifEntry 7 }

Syntax:
```
INTEGER {
    up(1),       -- ready to pass packets
    down(2),
    testing(3)   -- in some test mode
  }
```

Definition:
    The desired state of the interface. The testing(3) state
    indicates that no operational packets can be passed.

Access:
    read-write.

Status:
    mandatory.

OBJECT:
----

ifOperStatus { ifEntry 8 }

Syntax:
INTEGER {
up(1), -- ready to pass packets
down(2),
testing(3) -- in some test mode
}

Definition:
The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifLastChange { ifEntry 9 }

Syntax:
TimeTicks

Definition:
The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a zero value.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifInOctets { ifEntry 10 }

Syntax:
Counter

Definition:
The total number of octets received on the interface, including framing characters.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifInUcastPkts { ifEntry 11 }

Syntax:
Counter

Definition:
The number of (subnet) unicast packets delivered to a higher-layer protocol.

Access:
read-only.

Status:
mandatory.

OBJECT:

----
ifInNUcastPkts { ifEntry 12 }

Syntax:
Counter

Definition:
The number of non-unicast (i.e., subnet broadcast or subnet multicast) packets delivered to a higher-layer protocol.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifInDiscards { ifEntry 13 }

Syntax:
Counter

Definition:
The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer

protocol. One possible reason for discarding such a packet could be to free up buffer space.

Access:
read-only.

Status:
mandatory.

OBJECT:

----
ifInErrors { ifEntry 14 }

Syntax:
Counter

Definition:
The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifInUnknownProtos { ifEntry 15 }

Syntax:
Counter

Definition:
The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

Access:
read-only.

Status:
mandatory.

McCloghrie & Rose [Page 18]

RFC 1066 MIB August 1988

OBJECT:
----
ifOutOctets { ifEntry 16 }

Syntax:
Counter

Definition:
The total number of octets transmitted out of the interface, including framing characters.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifOutUcastPkts { ifEntry 17 }

Syntax:
Counter

Definition:
The total number of packets that higher-level protocols requested be transmitted to a subnet-unicast address, including those that were discarded or not sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifOutNUcastPkts { ifEntry 18 }

Syntax:
Counter

Definition:
The total number of packets that higher-level protocols requested be transmitted to a non-unicast (i.e., a subnet broadcast or subnet multicast) address, including those

that were discarded or not sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifOutDiscards { ifEntry 19 }

Syntax:
Counter

Definition:
The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifOutErrors { ifEntry 20 }

Syntax:
Counter

Definition:
The number of outbound packets that could not be transmitted because of errors.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ifOutQLen { ifEntry 21 }

Syntax:
Gauge

Definition:
The length of the output packet queue (in packets).

Access:
read-only.

Status:
mandatory.

5.3. The Address Translation Group

Implementation of the Address Translation group is mandatory for all systems.

The Address Translation group contains one table which is the union across all interfaces of the translation tables for converting a NetworkAddress (e.g., an IP address) into a subnetwork-specific address. For lack of a better term, this document refers to such a subnetwork-specific address as a "physical" address.

Examples of such translation tables are: for broadcast media where ARP is in use, the translation table is equivalent to the ARP cache; or, on an X.25 network where non-algorithmic translation to X.121 addresses is required, the translation table contains the NetworkAddress to X.121 address equivalences.

```
OBJECT:
-------
    atTable { at 1 }

Syntax:
    SEQUENCE OF AtEntry

Definition:
```

The Address Translation tables contain the NetworkAddress
to "physical" address equivalences. Some interfaces do
not use translation tables for determining address
equivalences (e.g., DDN-X.25 has an algorithmic method);
if all interfaces are of this type, then the Address
Translation table is empty, i.e., has zero entries.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
atEntry { atTable 1 }

Syntax:
```
AtEntry ::= SEQUENCE {
    atIfIndex
```



```
        INTEGER,
    atPhysAddress
        OCTET STRING,
    atNetAddress
        NetworkAddress
}
```

Definition:
Each entry contains one NetworkAddress to "physical"
address equivalence.

Access:
read-write.

Status:
mandatory.

We now consider the individual components of each Address
Translation table entry:

OBJECT:
----
atIfIndex { atEntry 1 }

Syntax:
INTEGER

Definition:
The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
atPhysAddress { atEntry 2 }

Syntax:
OCTET STRING

Definition:
The media-dependent "physical" address.

Access:
read-write.

Status:
mandatory.

OBJECT:

----
atNetAddress { atEntry 3 }

Syntax:
NetworkAddress

Definition:
The NetworkAddress (e.g., the IP address) corresponding to the media-dependent "physical" address.

Access:
read-write.

Status:
mandatory.

5.4. The IP Group

Implementation of the IP group is mandatory for all systems.

OBJECT:
----
    ipForwarding { ip 1 }

Syntax:
    INTEGER {
        gateway(1),   -- entity forwards datagrams
        host(2)   -- entity does NOT forward datagrams
    }

Definition:
    The indication of whether this entity is acting as an IP
    gateway in respect to the forwarding of datagrams
    received by, but not addressed to, this entity. IP
    gateways forward datagrams; Hosts do not (except those
    Source-Routed via the host).

Access:
    read-only.

Status:
    mandatory.

OBJECT:
----
    ipDefaultTTL { ip 2 }

Syntax:
    INTEGER

Definition:
    The default value inserted into the Time-To-Live field of
    the IP header of datagrams originated at this entity,
    whenever a TTL value is not supplied by the transport
    layer protocol.

Access:
    read-write.

Status:
    mandatory.

OBJECT:
----
ipInReceives { ip 3 }

Syntax:
Counter

Definition:
The total number of input datagrams received from interfaces, including those received in error.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipInHdrErrors { ip 4 }

Syntax:
Counter

Definition:
The number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipInAddrErrors { ip 5 }

Syntax:

Counter

Definition:
The number of input datagrams discarded because the IP address in their IP header's destination field was not a

McCloghrie & Rose [Page 26]

RFC 1066 MIB August 1988

valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP Gateways and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipForwDatagrams { ip 6 }

Syntax:
Counter

Definition:
The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities which do not act as IP Gateways, this counter will include only those packets which were Source-Routed via this entity, and the Source-Route option processing was successful.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipInUnknownProtos { ip 7 }

Syntax:
Counter

Definition:
The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipInDiscards { ip 8 }

Syntax:
Counter

Definition:
The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (e.g. for lack of buffer space). Note that this counter does not include any datagrams discarded while awaiting re-assembly.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipInDelivers { ip 9 }

Syntax:
Counter

Definition:
The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipOutRequests { ip 10 }

Syntax:
Counter

Definition:
The total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted in ipForwDatagrams.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipOutDiscards { ip 11 }

Syntax:
Counter

Definition:
The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space). Note that this counter would include datagrams counted in ipForwDatagrams if any such packets met this (discretionary) discard criterion.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipOutNoRoutes { ip 12 }

Syntax:
Counter

McCloghrie & Rose [Page 29]

RFC 1066 MIB August 1988

Definition:
The number of IP datagrams discarded because no route could be found to transmit them to their destination. Note that this counter includes any packets counted in ipForwDatagrams which meet this "no-route" criterion.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipReasmTimeout { ip 13 }

Syntax:
INTEGER

Definition:
The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipReasmReqds { ip 14 }

Syntax:
Counter

Definition:
The number of IP fragments received which needed to be reassembled at this entity.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipReasmOKs { ip 15 }

Syntax:
Counter

Definition:
The number of IP datagrams successfully re-assembled.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipReasmFails { ip 16 }

Syntax:
Counter

Definition:
The number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc).

Note that this is not necessarily a count of discarded IP fragments since some algorithms (notably RFC 815's) can lose track of the number of fragments by combining them as they are received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipFragOKs { ip 17 }

Syntax:
Counter

Definition:
The number of IP datagrams that have been successfully fragmented at this entity.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipFragFails { ip 18 }

Syntax:
Counter

Definition:
The number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be, e.g., because their "Don't Fragment" flag was set.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipFragCreates { ip 19 }

Syntax:
Counter

Definition:
The number of IP datagram fragments that have been generated as a result of fragmentation at this entity.

Access:

read-only.

Status:
mandatory.

5.4.1. The IP Address Table

The Ip Address table contains this entity's IP addressing information.

OBJECT:
----
ipAddrTable { ip 20 }

Syntax:
SEQUENCE OF IpAddrEntry

Definition:
The table of addressing information relevant to this entity's IP addresses.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipAddrEntry { ipAddrTable 1 }

Syntax:
```
IpAddrEntry ::= SEQUENCE {
    ipAdEntAddr
        IpAddress,
    ipAdEntIfIndex
        INTEGER,
    ipAdEntNetMask
```

IpAddress,
ipAdEntBcastAddr
INTEGER
}

Definition:
The addressing information for one of this entity's IP addresses.

Access:
read-only.

McCloghrie & Rose [Page 33]

RFC 1066 MIB August 1988

Status:
mandatory.

OBJECT:
----
ipAdEntAddr { ipAddrEntry 1 }

Syntax:
IpAddress

Definition:
The IP address to which this entry's addressing information pertains.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipAdEntIfIndex { ipAddrEntry 2 }

Syntax:

INTEGER

Definition:
The index value which uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipAdEntNetMask { ipAddrEntry 3 }

Syntax:
IpAddress

Definition:
The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipAdEntBcastAddr { ipAddrEntry 4 }

Syntax:
INTEGER

Definition:
The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry. For example, when the Internet standard all-ones broadcast address is used, the value will be 1.

Access:
read-only.

Status:
mandatory.

5.4.2. The IP Routing Table

The IP Routing Table contains an entry for each route presently known to this entity. Note that the action to be taken in response to a request to read a non-existent entry, is specific to the network management protocol being used.

OBJECT:
----
ipRoutingTable { ip 21 }

Syntax:
SEQUENCE OF IpRouteEntry

Definition:
This entity's IP Routing table.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
ipRouteEntry { ipRoutingTable 1 }

Syntax:
IpRouteEntry ::= SEQUENCE {
ipRouteDest
IpAddress,
ipRouteIfIndex
INTEGER,
ipRouteMetric1
INTEGER,
ipRouteMetric2
INTEGER,
ipRouteMetric3
INTEGER,
ipRouteMetric4
INTEGER,
ipRouteNextHop
IpAddress,
ipRouteType
INTEGER,
ipRouteProto
INTEGER,
ipRouteAge
INTEGER
}

Definition:
A route to a particular destination.

Access:
read-write.

McCloghrie & Rose [Page 36]

RFC 1066 MIB August 1988

Status:
mandatory.

We now consider the individual components of each route in the
IP Routing Table:

OBJECT:
----
    ipRouteDest { ipRouteEntry 1 }

Syntax:
    IpAddress

Definition:
    The destination IP address of this route. An entry with
    a value of 0.0.0.0 is considered a default route.
    Multiple such default routes can appear in the table, but
    access to such multiple entries is dependent on the
    table-access mechanisms defined by the network management
    protocol in use.

Access:
    read-write.

Status:
    mandatory.

OBJECT:
----
    ipRouteIfIndex { ipRouteEntry 2 }

Syntax:
    INTEGER

Definition:
    The index value which uniquely identifies the local
    interface through which the next hop of this route should
    be reached. The interface identified by a particular
    value of this index is the same interface as identified
    by the same value of ifIndex.

Access:
    read-write.

Status:
    mandatory.

OBJECT:
----
ipRouteMetric1 { ipRouteEntry 3 }

Syntax:
INTEGER

Definition:
The primary routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
ipRouteMetric2 { ipRouteEntry 4 }

Syntax:
INTEGER

Definition:
An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
ipRouteMetric3 { ipRouteEntry 5 }

Syntax:
INTEGER

Definition:
An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
ipRouteMetric4 { ipRouteEntry 6 }

Syntax:
INTEGER

Definition:
An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.

Access:
read-write.

Status:
mandatory.

OBJECT:
----

ipRouteNextHop { ipRouteEntry 7 }

Syntax:
IpAddress

Definition:
The IP address of the next hop of this route.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
ipRouteType { ipRouteEntry 8 }

Syntax:

```
INTEGER {
    other(1),        -- none of the following

    invalid(2),      -- an invalidated route

                     -- route to directly
    direct(3),       -- connected (sub-)network

                     -- route to a non-local
    remote(4),       -- host/network/sub-network
}
```

Definition:
The type of route.

Access:
read-write.

Status:
mandatory.

OBJECT:
----
ipRouteProto { ipRouteEntry 9 }

Syntax:
INTEGER {
other(1), -- none of the following

-- non-protocol information,
-- e.g., manually configured
local(2), -- entries

-- set via a network management
netmgmt(3), -- protocol

-- obtained via ICMP,
icmp(4), -- e.g., Redirect

-- the remaining values are
-- all gateway routing protocols
egp(5),

ggp(6),
hello(7),
rip(8),
is-is(9),
es-is(10),
ciscoIgrp(11),
bbnSpfIgp(12),
oigp(13)
}

Definition:
The routing mechanism via which this route was learned.
Inclusion of values for gateway routing protocols is not
intended to imply that hosts should support those
protocols.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
ipRouteAge { ipRouteEntry 10 }

Syntax:
INTEGER

Definition:
The number of seconds since this route was last updated or otherwise determined to be correct. Note that no semantics of "too old" can be implied except through knowledge of the routing protocol by which the route was learned.

Access:
read-write.

Status:
mandatory.

5.5. The ICMP Group

Implementation of the ICMP group is mandatory for all systems.

The ICMP group contains the ICMP input and output statistics.

Note that individual counters for ICMP message (sub-)codes have been omitted from this (version of the) MIB for simplicity.

OBJECT:

----

icmpInMsgs { icmp 1 }

Syntax:

Counter

Definition:

The total number of ICMP messages which the entity received. Note that this counter includes all those counted by icmpInErrors.

Access:

read-only.

Status:

mandatory.

OBJECT:

----

icmpInErrors { icmp 2 }

Syntax:

Counter

Definition:

The number of ICMP messages which the entity received but determined as having errors (bad ICMP checksums, bad length, etc.).

Access:

read-only.

Status:

mandatory.

McCloghrie & Rose [Page 42]

RFC 1066 MIB August 1988

OBJECT:

----

icmpInDestUnreachs { icmp 3 }

Syntax:
Counter

Definition:
The number of ICMP Destination Unreachable messages
received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInTimeExcds { icmp 4 }

Syntax:
Counter

Definition:
The number of ICMP Time Exceeded messages received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInParmProbs { icmp 5 }

Syntax:
Counter

Definition:
The number of ICMP Parameter Problem messages received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInSrcQuenchs { icmp 6 }

Syntax:
Counter

Definition:
The number of ICMP Source Quench messages received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInRedirects { icmp 7 }

Syntax:
Counter

Definition:
The number of ICMP Redirect messages received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInEchos { icmp 8 }

Syntax:

Counter

Definition:
The number of ICMP Echo (request) messages received.

McCloghrie & Rose [Page 44]

RFC 1066 MIB August 1988

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInEchoReps { icmp 9 }

Syntax:
Counter

Definition:
The number of ICMP Echo Reply messages received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInTimestamps { icmp 10 }

Syntax:
Counter

Definition:
The number of ICMP Timestamp (request) messages received.

Access:

read-only.

Status:
mandatory.

OBJECT:
----
icmpInTimestampReps { icmp 11 }

Syntax:
Counter

Definition:
The number of ICMP Timestamp Reply messages received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInAddrMasks { icmp 12 }

Syntax:
Counter

Definition:
The number of ICMP Address Mask Request messages received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpInAddrMaskReps { icmp 13 }

Syntax:
Counter

Definition:
The number of ICMP Address Mask Reply messages received.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutMsgs { icmp 14 }

Syntax:
Counter

Definition:
The total number of ICMP messages which this entity attempted to send. Note that this counter includes all those counted by icmpOutErrors.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutErrors { icmp 15 }

Syntax:
Counter

Definition:
The number of ICMP messages which this entity did not send due to problems discovered within ICMP such as a lack of buffers. This value should not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram. In some implementations there may be no types of error which contribute to this counter's value.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutDestUnreachs { icmp 16 }

Syntax:
Counter

Definition:
The number of ICMP Destination Unreachable messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutTimeExcds { icmp 17 }

Syntax:
Counter

Definition:
The number of ICMP Time Exceeded messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutParmProbs { icmp 18 }

Syntax:
Counter

Definition:
The number of ICMP Parameter Problem messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutSrcQuenchs { icmp 19 }

Syntax:
Counter

McCloghrie & Rose [Page 48]

RFC 1066 MIB August 1988

Definition:
The number of ICMP Source Quench messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutRedirects { icmp 20 }

Syntax:
Counter

Definition:
The number of ICMP Redirect messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutEchos { icmp 21 }

Syntax:
Counter

Definition:
The number of ICMP Echo (request) messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutEchoReps { icmp 22 }

Syntax:
Counter

Definition:
The number of ICMP Echo Reply messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutTimestamps { icmp 23 }

Syntax:
Counter

Definition:
The number of ICMP Timestamp (request) messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutTimestampReps { icmp 24 }

Syntax:
Counter

Definition:
The number of ICMP Timestamp Reply messages sent.

Access:
read-only.

Status:
mandatory.

McCloghrie & Rose [Page 50]

OBJECT:
----
icmpOutAddrMasks { icmp 25 }

Syntax:
Counter

Definition:
The number of ICMP Address Mask Request messages sent.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
icmpOutAddrMaskReps { icmp 26 }

Syntax:
Counter

Definition:
The number of ICMP Address Mask Reply messages sent.

Access:
read-only.

Status:
mandatory.

5.6. The TCP Group

Implementation of the TCP group is mandatory for all systems that implement the TCP protocol.

Note that instances of object types that represent information about a particular TCP connection are transient; they persist only as long as the connection in question.

```
OBJECT:
-------
    tcpRtoAlgorithm { tcp 1 }

Syntax:
    INTEGER {
        other(1),    -- none of the following
        constant(2), -- a constant rto
        rsre(3),     -- MIL-STD-1778, Appendix B
        vanj(4)      -- Van Jacobson's algorithm [11]
    }

Definition:
    The algorithm used to determine the timeout value used
    for retransmitting unacknowledged octets.

Access:
    read-only.
```

Status:
mandatory.

OBJECT:
----
tcpRtoMin { tcp 2 }

Syntax:
INTEGER

Definition:
The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission timeout. In particular, when the timeout algorithm is rsre(3), an object of this type has the semantics of the LBOUND quantity described in RFC 793.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpRtoMax { tcp 3 }

Syntax:
INTEGER

Definition:
The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to

determine the retransmission timeout. In particular,
when the timeout algorithm is rsre(3), an object of
this type has the semantics of the UBOUND quantity
described in RFC 793.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpMaxConn { tcp 4 }

Syntax:
INTEGER

Definition:
The limit on the total number of TCP connections the
entity can support. In entities where the maximum
number of connections is dynamic, this object should
contain the value "-1".

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpActiveOpens { tcp 5 }

Syntax:
Counter

Definition:
The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpPassiveOpens { tcp 6 }

Syntax:
Counter

Definition:
The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpAttemptFails { tcp 7 }

Syntax:
Counter

Definition:
The number of times TCP connections have made a direct transition to the CLOSED state from either the

SYN-SENT state or the SYN-RCVD state, plus the number
of times TCP connections have made a direct transition
to the LISTEN state from the SYN-RCVD state.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpEstabResets { tcp 8 }

Syntax:
Counter

Definition:
The number of times TCP connections have made a direct
transition to the CLOSED state from either the
ESTABLISHED state or the CLOSE-WAIT state.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpCurrEstab { tcp 9 }

Syntax:
Gauge

Definition:
The number of TCP connections for which the current
state is either ESTABLISHED or CLOSE-WAIT.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpInSegs { tcp 10 }

Syntax:
Counter

Definition:
The total number of segments received, including those received in error. This count includes segments received on currently established connections.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpOutSegs { tcp 11 }

Syntax:
Counter

Definition:
The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.

Access:
read-only.

Status:
mandatory.

OBJECT:

----
tcpRetransSegs { tcp 12 }

Syntax:
Counter

McCloghrie & Rose [Page 56]

RFC 1066 MIB August 1988

Definition:
The total number of segments retransmitted - that is,
the number of TCP segments transmitted containing one
or more previously transmitted octets.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpConnTable { tcp 13 }

Syntax:
SEQUENCE OF TcpConnEntry

Definition:
A table containing TCP connection-specific
information.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpConnEntry { tcpConnTable 1 }

Syntax:

```
TcpConnEntry ::= SEQUENCE {
    tcpConnState
        INTEGER,
    tcpConnLocalAddress
        IpAddress,
    tcpConnLocalPort
        INTEGER (0..65535),
    tcpConnRemAddress
        IpAddress,
    tcpConnRemPort
        INTEGER (0..65535)
}
```

McCloghrie & Rose [Page 57]

RFC 1066 MIB August 1988

Definition:
    Information about a particular current TCP connection.
    An object of this type is transient, in that it ceases
    to exist when (or soon after) the connection makes the
    transition to the CLOSED state.

Access:
    read-only.

Status:
    mandatory.

OBJECT:
----
    tcpConnState { tcpConnEntry 1 }

Syntax:

```
INTEGER {
    closed(1),
    listen(2),
    synSent(3),
    synReceived(4),
    established(5),
    finWait1(6),
    finWait2(7),
    closeWait(8),
```

lastAck(9),
closing(10),
timeWait(11)
}

Definition:
The state of this TCP connection.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpConnLocalAddress { tcpConnEntry 2 }

Syntax:
IpAddress

Definition:
The local IP address for this TCP connection.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpConnLocalPort { tcpConnEntry 3 }

Syntax:
INTEGER (0..65535)

Definition:
The local port number for this TCP connection.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpConnRemAddress { tcpConnEntry 4 }

Syntax:
IpAddress

Definition:
The remote IP address for this TCP connection.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
tcpConnRemPort { tcpConnEntry 5 }

McCloghrie & Rose [Page 59]

RFC 1066 MIB August 1988

Syntax:
INTEGER (0..65535)

Definition:
The remote port number for this TCP connection.

Access:
read-only.

Status:
mandatory.

McCloghrie & Rose [Page 60]

RFC 1066 MIB August 1988

5.7. The UDP Group

Implementation of the UDP group is mandatory for all systems which implement the UDP protocol.

OBJECT:
----
udpInDatagrams { udp 1 }

Syntax:
Counter

Definition:
The total number of UDP datagrams delivered to UDP users.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
udpNoPorts { udp 2 }

Syntax:
Counter

Definition:
The total number of received UDP datagrams for which there was no application at the destination port.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
udpInErrors { udp 3 }

Syntax:
Counter

Definition:
The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
udpOutDatagrams { udp 4 }

Syntax:
Counter

Definition:
The total number of UDP datagrams sent from this entity.

Access:
read-only.

Status:
mandatory.

McCloghrie & Rose [Page 62]

RFC 1066 MIB August 1988

5.8. The EGP Group

Implementation of the EGP group is mandatory for all systems which implement the EGP protocol.

OBJECT:
----
egpInMsgs { egp 1 }

Syntax:
Counter

Definition:
The number of EGP messages received without error.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
egpInErrors { egp 2 }

Syntax:
Counter

Definition:
The number of EGP messages received that proved to be in error.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
egpOutMsgs { egp 3 }

Syntax:
Counter

McCloghrie & Rose [Page 63]

RFC 1066 MIB August 1988

Definition:
The total number of locally generated EGP messages.

Access:
read-only.

Status:
mandatory.

OBJECT:
----
egpOutErrors { egp 4 }

Syntax:
Counter

Definition:
The number of locally generated EGP messages not sent
due to resource limitations within an EGP entity.

Access:
read-only.

Status:
mandatory.

5.8.1. The EGP Neighbor Table

The Egp Neighbor table contains information about this entity's EGP neighbors.

```
OBJECT:
-------
    egpNeighTable { egp 5 }

Syntax:
    SEQUENCE OF EgpNeighEntry

Definition:
    The EGP neighbor table.

Access:
    read-only.

Status:
    mandatory.
```



```
OBJECT:
-------
    egpNeighEntry { egpNeighTable 1 }

Syntax:
    EgpNeighEntry ::= SEQUENCE {
        egpNeighState
            INTEGER,
        egpNeighAddr
            IpAddress
    }

Definition:
    Information about this entity's relationship with a
    particular EGP neighbor.

Access:
    read-only.
```

Status:
    mandatory.

We now consider the individual components of each EGP neighbor entry:

OBJECT:
----
    egpNeighState { egpNeighEntry 1 }

Syntax:
    INTEGER {
        idle(1),
        acquisition(2),
        down(3),
        up(4),
        cease(5)
    }

Definition:
    The EGP state of the local system with respect to this entry's EGP neighbor. Each EGP state is represented by a value that is one greater than the numerical value associated with said state in RFC 904.

Access:
    read-only.

McCloghrie & Rose [Page 65]

RFC 1066 MIB August 1988

Status:
    mandatory.

OBJECT:
----
    egpNeighAddr { egpNeighEntry 2 }

Syntax:
    IpAddress

Definition:
The IP address of this entry's EGP neighbor.

Access:
read-only.

Status:
mandatory.

McCloghrie & Rose [Page 66]

RFC 1066 MIB August 1988

6. Definitions

RFC1066-MIB { iso org(3) dod(6) internet(1) mgmt(2) 1 }

```
DEFINITIONS ::= BEGIN

IMPORTS
    mgmt, OBJECT-TYPE, NetworkAddress, IpAddress,
    Counter, Gauge, TimeTicks
      FROM RFC1065-SMI;

 mib      OBJECT IDENTIFIER ::= { mgmt 1 }

 system     OBJECT IDENTIFIER ::= { mib 1 }
 interfaces OBJECT IDENTIFIER ::= { mib 2 }
 at       OBJECT IDENTIFIER ::= { mib 3 }
 ip       OBJECT IDENTIFIER ::= { mib 4 }
 icmp     OBJECT IDENTIFIER ::= { mib 5 }
 tcp      OBJECT IDENTIFIER ::= { mib 6 }
 udp      OBJECT IDENTIFIER ::= { mib 7 }
 egp      OBJECT IDENTIFIER ::= { mib 8 }

 -- object types

 -- the System group

 sysDescr OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     ::= { system 1 }

 sysObjectID OBJECT-TYPE
     SYNTAX  OBJECT IDENTIFIER
     ACCESS  read-only
     STATUS  mandatory
     ::= { system 2 }

 sysUpTime OBJECT-TYPE
     SYNTAX  TimeTicks
     ACCESS  read-only
     STATUS  mandatory
     ::= { system 3 }

 -- the Interfaces group

 ifNumber OBJECT-TYPE
     SYNTAX  INTEGER
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { interfaces 1 }

-- the Interfaces table

ifTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF IfEntry
        ACCESS  read-write
        STATUS  mandatory
        ::= { interfaces 2 }

ifEntry OBJECT-TYPE
        SYNTAX  IfEntry
        ACCESS  read-write
        STATUS  mandatory
        ::= { ifTable 1 }

IfEntry ::= SEQUENCE {
    ifIndex
        INTEGER,
    ifDescr
        OCTET STRING,
    ifType
        INTEGER,
    ifMtu
        INTEGER,
    ifSpeed
        Gauge,
    ifPhysAddress
        OCTET STRING,
    ifAdminStatus
        INTEGER,
    ifOperStatus
        INTEGER,
    ifLastChange
        TimeTicks,
    ifInOctets
        Counter,
    ifInUcastPkts
        Counter,
    ifInNUcastPkts
        Counter,
```

```
ifInDiscards
    Counter,
ifInErrors
    Counter,
ifInUnknownProtos
```



```
    Counter,
ifOutOctets
    Counter,
ifOutUcastPkts
    Counter,
ifOutNUcastPkts
    Counter,
ifOutDiscards
    Counter,
ifOutErrors
    Counter,
ifOutQLen
    Gauge
}

ifIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 1 }

ifDescr OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 2 }

ifType OBJECT-TYPE
        SYNTAX  INTEGER {
                other(1),      -- none of the following
                regular1822(2),
                hdh1822(3),
                ddn-x25(4),
                rfc877-x25(5),
                ethernet-csmacd(6),
```

```
            iso88023-csmacd(7),
            iso88024-tokenBus(8),
            iso88025-tokenRing(9),
            iso88026-man(10),
            starLan(11),
            proteon-10MBit(12),
            proteon-80MBit(13),
            hyperchannel(14),
            fddi(15),
            lapb(16),
            sdlc(17),
            t1-carrier(18),
            cept(19),
```



```
            basicIsdn(20),
            primaryIsdn(21),
                        -- proprietary serial
            propPointToPointSerial(22)
          }
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 3 }

  ifMtu OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 4 }

  ifSpeed OBJECT-TYPE
      SYNTAX  Gauge
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 5 }

  ifPhysAddress OBJECT-TYPE
      SYNTAX  OCTET STRING
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 6 }
```

```
ifAdminStatus OBJECT-TYPE
       SYNTAX  INTEGER {
              up(1),        -- ready to pass packets
              down(2),
              testing(3)    -- in some test mode
              }
       ACCESS  read-write
       STATUS  mandatory
       ::= { ifEntry 7 }

ifOperStatus OBJECT-TYPE
       SYNTAX  INTEGER {
              up(1),        -- ready to pass packets
              down(2),
              testing(3)    -- in some test mode
              }
       ACCESS  read-only
       STATUS  mandatory
       ::= { ifEntry 8 }

ifLastChange OBJECT-TYPE
```



```
       SYNTAX  TimeTicks
       ACCESS  read-only
       STATUS  mandatory
       ::= { ifEntry 9 }

ifInOctets OBJECT-TYPE
       SYNTAX  Counter
       ACCESS  read-only
       STATUS  mandatory
       ::= { ifEntry 10 }

ifInUcastPkts OBJECT-TYPE
       SYNTAX  Counter
       ACCESS  read-only
       STATUS  mandatory
       ::=  { ifEntry 11 }

ifInNUcastPkts OBJECT-TYPE
       SYNTAX  Counter
```

```
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 12 }

ifInDiscards OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 13 }

ifInErrors OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 14 }

ifInUnknownProtos OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 15 }

ifOutOctets OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 16 }

ifOutUcastPkts OBJECT-TYPE
```



```
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 17 }

ifOutNUcastPkts OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { ifEntry 18 }
```

```
ifOutDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 19 }

ifOutErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 20 }

ifOutQLen OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        ::= { ifEntry 21 }

-- the Address Translation group

atTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF AtEntry
        ACCESS  read-write
        STATUS  mandatory
        ::= { at 1 }

atEntry OBJECT-TYPE
        SYNTAX  AtEntry
        ACCESS  read-write
        STATUS  mandatory
        ::= { atTable 1 }

AtEntry ::= SEQUENCE {
    atIfIndex
        INTEGER,
    atPhysAddress
        OCTET STRING,
```



```
    atNetAddress
        NetworkAddress
}
```

```
atIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        ::= { atEntry 1 }

atPhysAddress OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        ::= { atEntry 2 }

atNetAddress OBJECT-TYPE
        SYNTAX  NetworkAddress
        ACCESS  read-write
        STATUS  mandatory
        ::= { atEntry 3 }

-- the IP group

ipForwarding OBJECT-TYPE
        SYNTAX  INTEGER {
      gateway(1), -- entity forwards datagrams
      host(2)     -- entity does NOT forward datagrams
                }
        ACCESS  read-only
        STATUS  mandatory
        ::= { ip 1 }

ipDefaultTTL OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        ::= { ip 2 }

ipInReceives OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { ip 3 }

ipInHdrErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
```

```
          STATUS  mandatory
          ::= { ip 4 }

      ipInAddrErrors OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 5 }

      ipForwDatagrams OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 6 }

      ipInUnknownProtos OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 7 }

      ipInDiscards OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 8 }

      ipInDelivers OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 9 }

      ipOutRequests OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 10 }

      ipOutDiscards OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
```

```
          ::= { ip 11 }

      ipOutNoRoutes OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
```



```
          STATUS  mandatory
          ::= { ip 12 }

      ipReasmTimeout OBJECT-TYPE
          SYNTAX  INTEGER
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 13 }

      ipReasmReqds OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 14 }

      ipReasmOKs OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 15 }

      ipReasmFails OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 16 }

      ipFragOKs OBJECT-TYPE
          SYNTAX  Counter
          ACCESS  read-only
          STATUS  mandatory
          ::= { ip 17 }

      ipFragFails OBJECT-TYPE
          SYNTAX  Counter
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { ip 18 }

    ipFragCreates OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { ip 19 }

    -- the IP Interface table

    ipAddrTable OBJECT-TYPE
```



```
        SYNTAX  SEQUENCE OF IpAddrEntry
        ACCESS  read-only
        STATUS  mandatory
        ::= { ip 20 }

    ipAddrEntry OBJECT-TYPE
        SYNTAX  IpAddrEntry
        ACCESS  read-only
        STATUS  mandatory
        ::= { ipAddrTable 1 }

    IpAddrEntry ::= SEQUENCE {
      ipAdEntAddr
        IpAddress,
      ipAdEntIfIndex
        INTEGER,
      ipAdEntNetMask
        IpAddress,
      ipAdEntBcastAddr
        INTEGER
    }

    ipAdEntAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::=  { ipAddrEntry 1 }
```

```
ipAdEntIfIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::=  { ipAddrEntry 2 }

ipAdEntNetMask OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::=  { ipAddrEntry 3 }

ipAdEntBcastAddr OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { ipAddrEntry 4 }

-- the IP Routing table
```

McCloghrie & Rose [Page 76]

RFC 1066 MIB August 1988

```
ipRoutingTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF IpRouteEntry
        ACCESS  read-write
        STATUS  mandatory
        ::= { ip 21 }

ipRouteEntry OBJECT-TYPE
        SYNTAX  IpRouteEntry
        ACCESS  read-write
        STATUS  mandatory
        ::= { ipRoutingTable 1 }

IpRouteEntry ::= SEQUENCE {
    ipRouteDest
        IpAddress,
    ipRouteIfIndex
        INTEGER,
    ipRouteMetric1
        INTEGER,
```

```
    ipRouteMetric2
        INTEGER,
    ipRouteMetric3
        INTEGER,
    ipRouteMetric4
        INTEGER,
    ipRouteNextHop
        IpAddress,
    ipRouteType
        INTEGER,
    ipRouteProto
        INTEGER,
    ipRouteAge
        INTEGER
}

ipRouteDest OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-write
        STATUS  mandatory
        ::= { ipRouteEntry 1 }

ipRouteIfIndex  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        ::= { ipRouteEntry 2 }

ipRouteMetric1 OBJECT-TYPE
```

McCloghrie & Rose [Page 77]

RFC 1066 MIB August 1988

```
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        ::= { ipRouteEntry 3 }

ipRouteMetric2 OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        ::= { ipRouteEntry 4 }
```

```
ipRouteMetric3 OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipRouteEntry 5 }

ipRouteMetric4 OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipRouteEntry 6 }

ipRouteNextHop OBJECT-TYPE
      SYNTAX  IpAddress
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipRouteEntry 7 }

ipRouteType OBJECT-TYPE
      SYNTAX  INTEGER {
        other(1),      -- none of the following

        invalid(2),    -- an invalidated route

                       -- route to directly
        direct(3),     -- connected (sub-)network

                       -- route to a non-local
        remote(4),     -- host/network/sub-network
          }
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipRouteEntry 8 }

ipRouteProto OBJECT-TYPE
      SYNTAX  INTEGER {
        other(1),     -- none of the following
```

McCloghrie & Rose [Page 78]

RFC 1066 MIB August 1988

```
                      -- non-protocol information
                      --   e.g., manually
        local(2),     --   configured entries
```

```
                    -- set via a network
        netmgmt(3),   --   management protocol

                    -- obtained via ICMP,
        icmp(4),      --   e.g., Redirect

                    -- the following are
                    -- gateway routing protocols
        egp(5),
        ggp(6),
        hello(7),
        rip(8),
        is-is(9),
        es-is(10),
        ciscoIgrp(11),
        bbnSpfIgp(12),
        oigp(13)
          }
      ACCESS  read-only
      STATUS  mandatory
      ::= { ipRouteEntry 9 }

ipRouteAge OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-write
      STATUS  mandatory
      ::= { ipRouteEntry 10 }

-- the ICMP group

icmpInMsgs OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { icmp 1 }

icmpInErrors OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { icmp 2 }

icmpInDestUnreachs OBJECT-TYPE
      SYNTAX  Counter
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 3 }

    icmpInTimeExcds OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 4 }

    icmpInParmProbs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 5 }

    icmpInSrcQuenchs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 6 }

    icmpInRedirects OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 7 }

    icmpInEchos OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 8 }

    icmpInEchoReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 9 }

    icmpInTimestamps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
```

```
         STATUS  mandatory
         ::= { icmp 10 }

     icmpInTimestampReps OBJECT-TYPE
         SYNTAX  Counter
```

McCloghrie & Rose [Page 80]


```
         ACCESS  read-only
         STATUS  mandatory
         ::= { icmp 11 }

     icmpInAddrMasks OBJECT-TYPE
         SYNTAX  Counter
         ACCESS  read-only
         STATUS  mandatory
         ::= { icmp 12 }

     icmpInAddrMaskReps OBJECT-TYPE
         SYNTAX  Counter
         ACCESS  read-only
         STATUS  mandatory
         ::= { icmp 13 }

     icmpOutMsgs OBJECT-TYPE
         SYNTAX  Counter
         ACCESS  read-only
         STATUS  mandatory
         ::= { icmp 14 }

     icmpOutErrors OBJECT-TYPE
         SYNTAX  Counter
         ACCESS  read-only
         STATUS  mandatory
         ::= { icmp 15 }

     icmpOutDestUnreachs OBJECT-TYPE
         SYNTAX  Counter
         ACCESS  read-only
         STATUS  mandatory
         ::= { icmp 16 }

     icmpOutTimeExcds OBJECT-TYPE
```

```
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 17 }

    icmpOutParmProbs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 18 }

    icmpOutSrcQuenchs OBJECT-TYPE
        SYNTAX  Counter
```



```
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 19 }

    icmpOutRedirects OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 20 }

    icmpOutEchos OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 21 }

    icmpOutEchoReps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 22 }

    icmpOutTimestamps OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { icmp 23 }
```

```
icmpOutTimestampReps OBJECT-TYPE
       SYNTAX  Counter
       ACCESS  read-only
       STATUS  mandatory
       ::= { icmp 24 }

icmpOutAddrMasks OBJECT-TYPE
       SYNTAX  Counter
       ACCESS  read-only
       STATUS  mandatory
       ::= { icmp 25 }

icmpOutAddrMaskReps OBJECT-TYPE
       SYNTAX  Counter
       ACCESS  read-only
       STATUS  mandatory
       ::= { icmp 26 }

-- the TCP group
```



```
tcpRtoAlgorithm OBJECT-TYPE
       SYNTAX  INTEGER {
       other(1),    -- none of the following
       constant(2), -- a constant rto
       rsre(3),     -- MIL-STD-1778, Appendix B
       vanj(4)      -- Van Jacobson's algorithm [11]
               }
       ACCESS  read-only
       STATUS  mandatory
       ::= { tcp 1 }

tcpRtoMin OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-only
       STATUS  mandatory
       ::= { tcp 2 }

tcpRtoMax OBJECT-TYPE
       SYNTAX  INTEGER
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 3 }

    tcpMaxConn OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 4 }

    tcpActiveOpens OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 5 }

    tcpPassiveOpens OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 6 }

    tcpAttemptFails OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 7 }

    tcpEstabResets OBJECT-TYPE
```



```
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 8 }

    tcpCurrEstab OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcp 9 }
```

```
tcpInSegs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcp 10 }

tcpOutSegs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcp 11 }

tcpRetransSegs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcp 12 }

-- the TCP connections table

tcpConnTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF TcpConnEntry
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
    SYNTAX  TcpConnEntry
    ACCESS  read-only
    STATUS  mandatory
    ::= { tcpConnTable 1 }

TcpConnEntry ::= SEQUENCE {
  tcpConnState
    INTEGER,
  tcpConnLocalAddress
    IpAddress,
```



```
  tcpConnLocalPort
    INTEGER (0..65535),
  tcpConnRemAddress
```

```
        IpAddress,
    tcpConnRemPort
        INTEGER (0..65535)
}

tcpConnState OBJECT-TYPE
        SYNTAX  INTEGER {
                    closed(1),
                    listen(2),
                    synSent(3),
                    synReceived(4),
                    established(5),
                    finWait1(6),
                    finWait2(7),
                    closeWait(8),
                    lastAck(9),
                    closing(10),
                    timeWait(11)
                }
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcpConnEntry 1 }

tcpConnLocalAddress OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcpConnEntry 2 }

tcpConnLocalPort OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE
        SYNTAX  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
```

```
        ::= { tcpConnEntry 5 }

    -- the UDP group

    udpInDatagrams OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { udp 1 }

    udpNoPorts OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { udp 2 }

    udpInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { udp 3 }

    udpOutDatagrams OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { udp 4 }

    -- the EGP group

    egpInMsgs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { egp 1 }

    egpInErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { egp 2 }

    egpOutMsgs OBJECT-TYPE
```

```
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { egp 3 }
```



```
    egpOutErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        ::= { egp 4 }

    -- the EGP Neighbor table

    egpNeighTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF EgpNeighEntry
        ACCESS  read-only
        STATUS  mandatory
        ::= { egp 5 }

    egpNeighEntry OBJECT-TYPE
        SYNTAX  EgpNeighEntry
        ACCESS  read-only
        STATUS  mandatory
        ::= { egpNeighTable 1 }

    EgpNeighEntry ::= SEQUENCE {
       egpNeighState
           INTEGER,
       egpNeighAddr
           IpAddress
    }

    egpNeighState OBJECT-TYPE
        SYNTAX  INTEGER {
                    idle(1),
                    acquisition(2),
                    down(3),
                    up(4),
                    cease(5)
                }
```

```
        ACCESS  read-only
        STATUS  mandatory
        ::= { egpNeighEntry 1 }

    egpNeighAddr OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
        ::= { egpNeighEntry 2 }

    END
```


7. Acknowledgements

The initial draft of this memo was heavily influenced by the the HEMS [5] and SNMP [6] MIBs.

Its final form is the result of the suggestions, the dicussions, and the compromises reached by the members of the IETF MIB working group:

Karl Auerbach, Epilogue Technology
K. Ramesh Babu, Excelan
Lawrence Besaw, Hewlett-Packard
Jeffrey D. Case, University of Tennessee at Knoxville
James R. Davin, Proteon
Mark S. Fedor, NYSERNet
Robb Foster, BBN
Phill Gross, The MITRE Corporation
Bent Torp Jensen, Convergent Technology
Lee Labarre, The MITRE Corporation
Dan Lynch, Advanced Computing Environments
Keith McCloghrie, The Wollongong Group
Dave Mackie, 3Com/Bridge
Craig Partridge, BBN (chair)
Jim Robertson, 3Com/Bridge
Marshall T. Rose, The Wollongong Group
Greg Satz, cisco
Martin Lee Schoffstall, Rensselaer Polytechnic Institute
Lou Steinberg, IBM

Dean Throop, Data General
Unni Warrier, Unisys

8. References

[1] Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards", RFC 1052, IAB, April 1988.

[2] Information processing systems - Open Systems Interconnection, "Management Information Services Definition", International Organization for Standardization, Draft Proposal 9595/2, December 1987.

[3] Information processing systems - Open Systems Interconnection, "Management Information Protocol Specification", International Organization for Standardization, Draft Proposal 9596/2, December 1987.

[4] Rose M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", RFC 1065, TWG, August 1988.

[5] Partridge C., and G. Trewitt, "The High-Level Entity Management System (HEMS)", RFCs 1021-1024, BBN and Stanford, October 1987.

[6] Case, J., M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol", RFC 1067, University of Tennessee At Knoxville, NYSERNet, Rensselaer Polytechnic, Proteon, August 1988.

[7] LaBarre, L., "Structure and Identification of Management Information for the Internet", Internet Engineering Task Force working note, Network Information Center, SRI International, Menlo Park, California, April 1988.

[8] LaBarre, L., "Transport Layer Management Information: TCP", Internet Engineering Task Force working note in preparation. Network Information Center, SRI International, Menlo Park, California, (unpublished).

[9] Information processing systems - Open Systems Interconnection, "Specification of Abstract Syntax Notation One (ASN.1)", International Organization for Standardization, International Standard 8824, December 1987.

[10] Information processing systems - Open Systems Interconnection, "Specification of Basic Encoding Rules for Abstract Notation One (ASN.1)", International Organization for Standardization, International Standard 8825, December 1987.

[11] Jacobson, V., "Congestion Avoidance and Control", SIGCOMM, 1988,

Stanford, California.

McCloghrie & Rose [Page 90]

Found in path(s):

* /opt/cola/permits/1601387782_1679299635.2639437/0/net-snmp-2-9-1-tgz/package/ref/rfc/v1/rfc1066.txt

1.13 glibc 2.28

1.13.1 Available under license :

/* _setjmp is implemented in setjmp.S */
@c The GNU Lesser General Public License.
@center Version 2.1, February 1999

@c This file is intended to be included within another document,
@c hence no sectioning command or @node.

@display
Copyright @copyright{} 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.]
@end display

@subheading 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---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 @dfn{Lesser} General Public License because it does @emph{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.

@subheading TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

@enumerate 0
@item
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.

@item
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.

@item
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:

@enumerate a
@item
The modified work must itself be a software library.

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

@item
You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

@item
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.)
@end enumerate

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.

@item
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.

@item
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.

@item
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.

@item
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:

@enumerate a
@item
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.)

@item
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.

@item
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.

@item
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.

@item
Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.
@end enumerate

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.

@item
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:

@enumerate a
@item
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.

@item
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.
@end enumerate

@item
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.

@item
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.

@item
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.

@item
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.

@item
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.

@item
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.

@item
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.

@center @b{NO WARRANTY}

@item

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.

@item
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 enumerate

@subheading END OF TERMS AND CONDITIONS

@page
@subheading 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.

@smallexample
@var{one line to give the library's name and an idea of what it does.}
Copyright (C) @var{year} @var{name of author}

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

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

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

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:

@smallexample
Yoyodyne, Inc., hereby disclaims all copyright interest in the library
`Frob' (a library for tweaking knobs) written by James Random Hacker.

@var{signature of Ty Coon}, 1 April 1990
Ty Coon, President of Vice
@end smallexample

That's all there is to it!

/* setjmp is implemented in setjmp.S */

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

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

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

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

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

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

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

a) The modified work must itself be a software library.

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

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

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

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

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

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

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

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the

Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

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

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

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

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent

license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

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

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

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

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

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

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

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free

Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

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

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

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public

License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

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

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

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
This file contains the copying permission notices for various files in the
GNU C Library distribution that have copyright owners other than the Free
Software Foundation. These notices all require that a copy of the notice
be included in the accompanying documentation and be distributed with
binary distributions of the code, so be sure to include this file along
with any binary distributions derived from the GNU C Library.

All code incorporated from 4.4 BSD is distributed under the following
license:

Copyright (C) 1991 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.

The DNS resolver code, taken from BIND 4.9.5, is copyrighted by UC
Berkeley, by Digital Equipment Corporation and by Internet Software
Consortium. The DEC portions are under the following license:

The ISC portions are under the following license:

ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM 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.

The Sun RPC support (from rpcsrc-4.0) is covered by the following license:

Copyright (c) 2010, Oracle America, Inc.

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 "Oracle America, 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 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 following CMU license covers some of the support code for Mach, derived from Mach 3.0:

Mach Operating System
Copyright (C) 1991,1990,1989 Carnegie Mellon University
All Rights Reserved.

The file if_ppp.h is under the following CMU license:

The following license covers the files from Intel's "Highly Optimized Mathematical Functions for Itanium" collection:

Intel License Agreement

The files inet/getnameinfo.c and sysdeps/posix/getaddrinfo.c are copyright (C) by Craig Metz and are distributed under the following license:

/* The Inner Net License, Version 2.00

way or form.
1. All terms of the 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. [The copyright holder has authorized the removal of this clause.]
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.

If these license terms cause you a real problem, contact the author. */

The file sunrpc/des_impl.c is copyright Eric Young:

Copyright (C) 1992 Eric Young
Collected from libdes and modified for SECURE RPC by Martin Kuck 1994
This file is distributed under the terms of the GNU Lesser General Public License, version 2.1 or later - see the file COPYING.LIB for details.
If you did not receive a copy of the license with this program, please see <http://www.gnu.org/licenses/> to obtain a copy.

The file inet/rcmd.c is under a UCB copyright and the following:

Copyright (C) 1998 WIDE Project.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the project nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PROJECT 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 PROJECT 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 file posix/runtests.c is copyright Tom Lord:

Copyright 1995 by Tom Lord

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the copyright holder not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

Tom Lord DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL TOM LORD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The posix/rxspencer tests are copyright Henry Spencer:

Copyright 1992, 1993, 1994, 1997 Henry Spencer. All rights reserved.
This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this
  software, no matter how awful, even if they arise from flaws in it.

2. The origin of this software must not be misrepresented, either by
  explicit claim or by omission. Since few users ever read sources,
  credits must appear in the documentation.

3. Altered versions must be plainly marked as such, and must not be
  misrepresented as being the original software. Since few users
  ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

The file posix/PCRE.tests is copyright University of Cambridge:

Copyright (c) 1997-2003 University of Cambridge

Permission is granted to anyone to use this software for any purpose on any computer system, and to redistribute it freely, subject to the following restrictions:

1. This software 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.

2. The origin of this software must not be misrepresented, either by
  explicit claim or by omission. In practice, this means that if you use
  PCRE in software that you distribute to others, commercially or
  otherwise, you must put a sentence like this

    Regular expression support is provided by the PCRE library package,
    which is open source software, written by Philip Hazel, and copyright
    by the University of Cambridge, England.

  somewhere reasonably visible in your documentation and in any relevant
  files or online help data or similar. A reference to the ftp site for
  the source, that is, to

    ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/

  should also be given in the documentation. However, this condition is not
  intended to apply to whole chains of software. If package A includes PCRE,
  it must acknowledge it, but if package B is software that includes package
  A, the condition is not imposed on package B (unless it uses PCRE
  independently).

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

4. If PCRE is embedded in any software that is released under the GNU
General Purpose Licence (GPL), or Lesser General Purpose Licence (LGPL),
then the terms of that licence shall supersede any condition above with
which it is incompatible.

Files from Sun fdlibm are copyright Sun Microsystems, Inc.:

Copyright (C) 1993 by Sun Microsystems, Inc. All rights reserved.

Developed at SunPro, a Sun Microsystems, Inc. business.
Permission to use, copy, modify, and distribute this
software is freely granted, provided that this notice
is preserved.

Various long double libm functions are copyright Stephen L. Moshier:

Copyright 2001 by Stephen L. Moshier <moshier@na-net.ornl.gov>

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, see
<http://www.gnu.org/licenses/>. */
UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories
http://www.unicode.org/Public/, http://www.unicode.org/reports/, and
http://www.unicode.org/cldr/data/. Unicode Data Files do not include PDF
online code charts under the directory http://www.unicode.org/Public/.
Software includes any source code published in the Unicode Standard or under
the directories http://www.unicode.org/Public/,
http://www.unicode.org/reports/, and http://www.unicode.org/cldr/data/.

NOTICE TO USER: Carefully read the following legal agreement. BY
DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES
("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND
AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF
YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA
FILES OR SOFTWARE.

@enumerate 0
@item
PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document @dfn{free} in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

@item
APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent'' copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ``Transparent'' is called ``Opaque''.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, La@TeX{} input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The ``Title Page'' means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ``Title Page'' means

the text near the most prominent appearance of the work's title,
preceding the beginning of the body of the text.

The ``publisher'' means any person or entity that distributes copies
of the Document to the public.

A section ``Entitled XYZ'' means a named subunit of the Document whose
title either is precisely XYZ or contains XYZ in parentheses following
text that translates XYZ in another language. (Here XYZ stands for a
specific section name mentioned below, such as ``Acknowledgements'',
``Dedications'', ``Endorsements'', or ``History''.) To ``Preserve the Title''
of such a section when you modify the Document means that it remains a
section ``Entitled XYZ'' according to this definition.

The Document may include Warranty Disclaimers next to the notice which
states that this License applies to the Document. These Warranty
Disclaimers are considered to be included by reference in this
License, but only as regards disclaiming warranties: any other
implication that these Warranty Disclaimers may have is void and has
no effect on the meaning of this License.

@item
VERBATIM COPYING

You may copy and distribute the Document in any medium, either
commercially or noncommercially, provided that this License, the
copyright notices, and the license notice saying this License applies
to the Document are reproduced in all copies, and that you add no other
conditions whatsoever to those of this License. You may not use
technical measures to obstruct or control the reading or further
copying of the copies you make or distribute. However, you may accept
compensation in exchange for copies. If you distribute a large enough
number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and
you may publicly display copies.

@item
COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have
printed covers) of the Document, numbering more than 100, and the
Document's license notice requires Cover Texts, you must enclose the
copies in covers that carry, clearly and legibly, all these Cover
Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on
the back cover. Both covers must also clearly and legibly identify
you as the publisher of these copies. The front cover must present
the full title with all words of the title equally prominent and

visible. You may add other material on the covers in addition.
Copying with changes limited to the covers, as long as they preserve
the title of the Document and satisfy these conditions, can be treated
as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit
legibly, you should put the first ones listed (as many as fit
reasonably) on the actual cover, and continue the rest onto adjacent
pages.

If you publish or distribute Opaque copies of the Document numbering
more than 100, you must either include a machine-readable Transparent
copy along with each Opaque copy, or state in or with each Opaque copy
a computer-network location from which the general network-using
public has access to download using public-standard network protocols
a complete Transparent copy of the Document, free of added material.
If you use the latter option, you must take reasonably prudent steps,
when you begin distribution of Opaque copies in quantity, to ensure
that this Transparent copy will remain thus accessible at the stated
location until at least one year after the last time you distribute an
Opaque copy (directly or through your agents or retailers) of that
edition to the public.

It is requested, but not required, that you contact the authors of the
Document well before redistributing any large number of copies, to give
them a chance to provide you with an updated version of the Document.

@item
MODIFICATIONS

You may copy and distribute a Modified Version of the Document under
the conditions of sections 2 and 3 above, provided that you release
the Modified Version under precisely this License, with the Modified
Version filling the role of the Document, thus licensing distribution
and modification of the Modified Version to whoever possesses a copy
of it. In addition, you must do these things in the Modified Version:

@enumerate A
@item
Use in the Title Page (and on the covers, if any) a title distinct
from that of the Document, and from those of previous versions
(which should, if there were any, be listed in the History section
of the Document). You may use the same title as a previous version
if the original publisher of that version gives permission.

@item
List on the Title Page, as authors, one or more persons or entities
responsible for authorship of the modifications in the Modified

Version, together with at least five of the principal authors of the
Document (all of its principal authors, if it has fewer than five),
unless they release you from this requirement.

@item
State on the Title page the name of the publisher of the
Modified Version, as the publisher.

@item
Preserve all the copyright notices of the Document.

@item
Add an appropriate copyright notice for your modifications
adjacent to the other copyright notices.

@item
Include, immediately after the copyright notices, a license notice
giving the public permission to use the Modified Version under the
terms of this License, in the form shown in the Addendum below.

@item
Preserve in that license notice the full lists of Invariant Sections
and required Cover Texts given in the Document's license notice.

@item
Include an unaltered copy of this License.

@item
Preserve the section Entitled ``History'', Preserve its Title, and add
to it an item stating at least the title, year, new authors, and
publisher of the Modified Version as given on the Title Page. If
there is no section Entitled ``History'' in the Document, create one
stating the title, year, authors, and publisher of the Document as
given on its Title Page, then add an item describing the Modified
Version as stated in the previous sentence.

@item
Preserve the network location, if any, given in the Document for
public access to a Transparent copy of the Document, and likewise
the network locations given in the Document for previous versions
it was based on. These may be placed in the ``History'' section.
You may omit a network location for a work that was published at
least four years before the Document itself, or if the original
publisher of the version it refers to gives permission.

@item
For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve
the Title of the section, and preserve in the section all the

substance and tone of each of the contributor acknowledgements and/or
dedications given therein.

@item
Preserve all the Invariant Sections of the Document,
unaltered in their text and in their titles. Section numbers
or the equivalent are not considered part of the section titles.

@item
Delete any section Entitled ``Endorsements''. Such a section
may not be included in the Modified Version.

@item
Do not retitle any existing section to be Entitled ``Endorsements'' or
to conflict in title with any Invariant Section.

@item
Preserve any Warranty Disclaimers.
@end enumerate

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document, you may at your option designate some or all
of these sections as invariant. To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements'', provided it contains
nothing but endorsements of your Modified Version by various
parties---for example, statements of peer review or that the text has
been approved by an organization as the authoritative definition of a
standard.

You may add a passage of up to five words as a Front-Cover Text, and a
passage of up to 25 words as a Back-Cover Text, to the end of the list
of Cover Texts in the Modified Version. Only one passage of
Front-Cover Text and one of Back-Cover Text may be added by (or
through arrangements made by) any one entity. If the Document already
includes a cover text for the same cover, previously added by you or
by arrangement made by the same entity you are acting on behalf of,
you may not add another; but you may replace the old one, on explicit
permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License
give permission to use their names for publicity for or to assert or
imply endorsement of any Modified Version.

@item

COMBINING DOCUMENTS

You may combine the Document with other documents released under this
License, under the terms defined in section 4 above for modified
versions, provided that you include in the combination all of the
Invariant Sections of all of the original documents, unmodified, and
list them all as Invariant Sections of your combined work in its
license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and
multiple identical Invariant Sections may be replaced with a single
copy. If there are multiple Invariant Sections with the same name but
different contents, make the title of each such section unique by
adding at the end of it, in parentheses, the name of the original
author or publisher of that section if known, or else a unique number.
Make the same adjustment to the section titles in the list of
Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History''
in the various original documents, forming one section Entitled
``History''; likewise combine any sections Entitled ``Acknowledgements'',
and any sections Entitled ``Dedications''. You must delete all
sections Entitled ``Endorsements.''

@item
COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents
released under this License, and replace the individual copies of this
License in the various documents with a single copy that is included in
the collection, provided that you follow the rules of this License for
verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute
it individually under this License, provided you insert a copy of this
License into the extracted document, and follow this License in all
other respects regarding verbatim copying of that document.

@item
AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate
and independent documents or works, in or on a volume of a storage or
distribution medium, is called an ``aggregate'' if the copyright
resulting from the compilation is not used to limit the legal rights
of the compilation's users beyond what the individual works permit.
When the Document is included in an aggregate, this License does not
apply to the other works in the aggregate which are not themselves

derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item
TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'', ``Dedications'', or ``History'', the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

@item
TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

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, receipt of a copy of some or all of the same material does
not give you any rights to use it.

@item
FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions
of the GNU Free Documentation 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. See
@uref{http://www.gnu.org/copyleft/}.

Each version of the License is given a distinguishing version number.
If the Document specifies that a particular numbered version of this
License ``or any later version'' applies to it, you have the option of
following the terms and conditions either of that specified version or
of any later version that has been published (not as a draft) by the
Free Software Foundation. If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation. If the Document
specifies that a proxy can decide which future versions of this
License can be used, that proxy's public statement of acceptance of a
version permanently authorizes you to choose that version for the
Document.

@item
RELICENSING

``Massive Multiauthor Collaboration Site'' (or ``MMC Site'') means any
World Wide Web server that publishes copyrightable works and also
provides prominent facilities for anybody to edit those works. A
public wiki that anybody can edit is an example of such a server. A
``Massive Multiauthor Collaboration'' (or ``MMC'') contained in the
site means any set of copyrightable works thus published on the MMC
site.

``CC-BY-SA'' means the Creative Commons Attribution-Share Alike 3.0
license published by Creative Commons Corporation, a not-for-profit
corporation with a principal place of business in San Francisco,
California, as well as future copyleft versions of that license

published by that same organization.

``Incorporate'' means to publish or republish a Document, in whole or
in part, as part of another Document.

An MMC is ``eligible for relicensing'' if it is licensed under this
License, and if all works that were first published under this License
somewhere other than this MMC, and subsequently incorporated in whole
or in part into the MMC, (1) had no cover texts or invariant sections,
and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site
under CC-BY-SA on the same site at any time before August 1, 2009,
provided the MMC is eligible for relicensing.

@end enumerate

@page
@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and
license notices just after the title page:

```
@smallexample
@group
  Copyright (C)  @var{year}  @var{your name}.
  Permission is granted to copy, distribute and/or modify this document
  under the terms of the GNU Free Documentation License, Version 1.3
  or any later version published by the Free Software Foundation;
  with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
  Texts.  A copy of the license is included in the section entitled ``GNU
  Free Documentation License''.
@end group
@end smallexample
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the ``with@dots{}Texts.''@: line with this:

```
@smallexample
@group
    with the Invariant Sections being @var{list their titles}, with
    the Front-Cover Texts being @var{list}, and with the Back-Cover Texts
    being @var{list}.
@end group
@end smallexample
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

@c Local Variables:
@c ispell-local-pdict: "ispell-dict"
@c End:

1.14 linux-kernel 5.4.74

1.14.1 Available under license :

FlashPoint Driver Developer's Kit
Version 1.0

This program is free software; you may redistribute and/or modify it under the terms of either:

a) the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version,

or

b) the "BSD-style License" included below.

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 either the GNU General Public License or the BSD-style License below 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., 675 Mass Ave, Cambridge, MA 02139, USA.

The BSD-style License is as follows:

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 this LICENSE.FlashPoint

file, without modification, this list of conditions, and the following
disclaimer. The following copyright notice must appear immediately at
the beginning of all source files:

Copyright 1995-1996 by Mylex Corporation. All Rights Reserved

This file is available under both the GNU General Public License
and a BSD-style copyright; see LICENSE.FlashPoint for details.

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 Mylex Corporation may not be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY MYLEX CORP. ``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.

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 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 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 Library General
Public License instead of this License.
Copyright (c) 2009-2013 QLogic Corporation
QLogic Linux qlcnic NIC Driver

You may modify and redistribute the device driver code under the
GNU General Public License (a copy of which is attached hereto as
Exhibit A) published by the Free Software Foundation (version 2).

EXHIBIT A

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.
Valid-License-Identifier: BSD-3-Clause
SPDX-URL: https://spdx.org/licenses/BSD-3-Clause.html
Usage-Guide:
To use the BSD 3-clause "New" or "Revised" License put the following SPDX
tag/value pair into a comment according to the placement guidelines in
the licensing rules documentation:
SPDX-License-Identifier: BSD-3-Clause
License-Text:

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.

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.

Valid-License-Identifier: ISC
SPDX-URL: https://spdx.org/licenses/ISC.html
Usage-Guide:
To use the ISC License put the following SPDX tag/value pair into a
comment according to the placement guidelines in the licensing rules
documentation:
SPDX-License-Identifier: ISC
License-Text:

ISC License

Copyright (c) <year> <copyright holders>

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.

Valid-License-Identifier: X11
SPDX-URL: https://spdx.org/licenses/X11.html
Usage-Guide:
To use the X11 put the following SPDX tag/value pair into a comment

according to the placement guidelines in the licensing rules
documentation:
SPDX-License-Identifier: X11
License-Text:

X11 License

Copyright (C) 1996 X Consortium

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 X CONSORTIUM 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 the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

X Window System is a trademark of X Consortium, Inc.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software

Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed

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

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

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

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

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

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

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

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

the Program is not required to print an announcement.)

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

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

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

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

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

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

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

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

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

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

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

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

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

refrain entirely from distribution of the Program.

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

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

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

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

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

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

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

of promoting the sharing and reuse of software generally.

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

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

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

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

GNU General Public License for more details.

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

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

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

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

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

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

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

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

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General
Public License instead of this License.
Copyright (c) 2003-2013 QLogic Corporation
QLogic Linux iSCSI Driver

This program includes a device driver for Linux 3.x.
You may modify and redistribute the device driver code under the
GNU General Public License (a copy of which is attached hereto as
Exhibit A) published by the Free Software Foundation (version 2).

EXHIBIT A

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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.

/* nicstar.c v0.22 Jawaid Bazyar (bazyar@hypermall.com)
* nicstar.c, M. Welsh (matt.welsh@cl.cam.ac.uk)
*
* Hacked October, 1997 by Jawaid Bazyar, Interlink Advertising Services Inc.
* http://www.hypermall.com/
* 10/1/97 - commented out CFG_PHYIE bit - we don't care when the PHY
* interrupts us (except possibly for removal/insertion of the cable?)
* 10/4/97 - began heavy inline documentation of the code. Corrected typos
* and spelling mistakes.
* 10/5/97 - added code to handle PHY interrupts, disable PHY on

* loss of link, and correctly re-enable PHY when link is
* re-established. (put back CFG_PHYIE)
*
* Modified to work with the IDT7721 nicstar -- AAL5 (tested) only.
*
* R. D. Rechenmacher <ron@fnal.gov>, Aug. 6, 1997
*
* Linux driver for the IDT77201 NICStAR PCI ATM controller.
* PHY component is expected to be 155 Mbps S/UNI-Lite or IDT 77155;
* see init_nicstar() for PHY initialization to change this. This driver
* expects the Linux ATM stack to support scatter-gather lists
* (skb->atm.iovcnt != 0) for Rx skb's passed to vcc->push.
*
* Implementing minimal-copy of received data:
* IDT always receives data into a small buffer, then large buffers
* as needed. This means that data must always be copied to create
* the linear buffer needed by most non-ATM protocol stacks (e.g. IP)
* Fix is simple: make large buffers large enough to hold entire
* SDU, and leave <small_buffer_data> bytes empty at the start. Then
* copy small buffer contents to head of large buffer.
* Trick is to avoid fragmenting Linux, due to need for a lot of large
* buffers. This is done by 2 things:
* 1) skb->destructor / skb->atm.recycle_buffer
* combined, allow nicstar_free_rx_skb to be called to
* recycle large data buffers
* 2) skb_clone of received buffers
* See nicstar_free_rx_skb and linearize_buffer for implementation
* details.
*
*
*
* Copyright (c) 1996 University of Cambridge Computer Laboratory
*
* 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., 675 Mass Ave, Cambridge, MA 02139, USA.
*
* M. Welsh, 6 July 1996

*
*
*/
Code in this directory written at the IDA Supercomputing Research Center carries the following copyright and license.

QLogic Linux Networking HBA Driver

This program includes a device driver for Linux 2.6 that may be distributed with QLogic hardware specific firmware binary file. You may modify and redistribute the device driver code under the GNU General Public License as published by the Free Software Foundation (version 2 or a later version).

You may redistribute the hardware specific firmware binary file under the following terms:

1. Redistribution of source code (only if applicable), must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistribution 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 QLogic Corporation may not be used to
endorse or promote products derived from this software
without specific prior written permission

REGARDLESS OF WHAT LICENSING MECHANISM IS USED OR APPLICABLE, THIS PROGRAM IS PROVIDED BY QLOGIC CORPORATION "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.

USER ACKNOWLEDGES AND AGREES THAT USE OF THIS PROGRAM WILL NOT CREATE OR GIVE GROUNDS FOR A LICENSE BY IMPLICATION, ESTOPPEL, OR OTHERWISE IN ANY INTELLECTUAL PROPERTY RIGHTS (PATENT, COPYRIGHT, TRADE SECRET, MASK WORK, OR OTHER PROPRIETARY RIGHT) EMBODIED IN ANY OTHER QLOGIC HARDWARE OR SOFTWARE EITHER SOLELY OR IN COMBINATION WITH THIS PROGRAM.
The files in this directory and elsewhere which refer to this LICENCE file are part of JFFS2, the Journalling Flash File System v2.

Copyright 2001-2007 Red Hat, Inc. and others

JFFS2 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 or (at your option) any later version.

JFFS2 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 JFFS2; 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 templates or use
macros or inline functions from these files, or you compile these
files and link them with other works to produce a work based on these
files, these files do not by themselves cause the resulting work to be
covered by the GNU General Public License. However the source code for
these files must still be made available in accordance with section (3)
of the GNU General Public License.

This exception does not invalidate any other reasons why a work based on
this file might be covered by the GNU General Public License.
The Linux Kernel is provided under:

SPDX-License-Identifier: GPL-2.0 WITH Linux-syscall-note

Being under the terms of the GNU General Public License version 2 only,
according with:

LICENSES/preferred/GPL-2.0

With an explicit syscall exception, as stated at:

LICENSES/exceptions/Linux-syscall-note

In addition, other licenses may also apply. Please see:

Documentation/process/license-rules.rst

for more details.
Valid-License-Identifier: MIT
SPDX-URL: https://spdx.org/licenses/MIT.html
Usage-Guide:
To use the MIT License put the following SPDX tag/value pair into a
comment according to the placement guidelines in the licensing rules
documentation:
SPDX-License-Identifier: MIT
License-Text:

MIT License

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.

Valid-License-Identifier: BSD-2-Clause
SPDX-URL: https://spdx.org/licenses/BSD-2-Clause.html
Usage-Guide:
 To use the BSD 2-clause "Simplified" License put the following SPDX
 tag/value pair into a comment according to the placement guidelines in
 the licensing rules documentation:
   SPDX-License-Identifier: BSD-2-Clause
License-Text:

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.

SPDX-Exception-Identifier: Linux-syscall-note
SPDX-URL: https://spdx.org/licenses/Linux-syscall-note.html
SPDX-Licenses: GPL-2.0, GPL-2.0+, GPL-1.0+, LGPL-2.0, LGPL-2.0+, LGPL-2.1, LGPL-2.1+, GPL-2.0-only, GPL-2.0-or-later
Usage-Guide:

This exception is used together with one of the above SPDX-Licenses
to mark user space API (uapi) header files so they can be included
into non GPL compliant user space application code.
To use this exception add it with the keyword WITH to one of the
identifiers in the SPDX-Licenses tag:
SPDX-License-Identifier: <SPDX-License> WITH Linux-syscall-note
License-Text:

NOTE! This copyright does *not* cover user programs that use kernel
services by normal system calls - this is merely considered normal use
of the kernel, and does *not* fall under the heading of "derived work".
Also note that the GPL below is copyrighted by the Free Software
Foundation, but the instance of code that it refers to (the Linux
kernel) is copyrighted by me and others who actually wrote it.

Also note that the only valid version of the GPL as far as the kernel
is concerned is _this_ particular version of the license (ie v2, not
v2.2 or v3.x or whatever), unless explicitly otherwise stated.

Linus Torvalds
Valid-License-Identifier: BSD-3-Clause-Clear
SPDX-URL: https://spdx.org/licenses/BSD-3-Clause-Clear.html
Usage-Guide:
To use the BSD 3-clause "Clear" License put the following SPDX
tag/value pair into a comment according to the placement guidelines in
the licensing rules documentation:
SPDX-License-Identifier: BSD-3-Clause-Clear
License-Text:

The Clear BSD License

Copyright (c) [xxxx]-[xxxx] [Owner Organization]
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted (subject to the limitations in the disclaimer
below) 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 [Owner Organization] nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

QLogic Linux FC-FCoE Driver

This program includes a device driver for Linux 3.x.
You may modify and redistribute the device driver code under the
GNU General Public License (a copy of which is attached hereto as
Exhibit A) published by the Free Software Foundation (version 2).

EXHIBIT A

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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.

This is the full-colour version of the currently unofficial Linux logo ("currently unofficial" just means that there has been no paperwork and that I have not really announced it yet). It was created by Larry Ewing, and is freely usable as long as you acknowledge Larry as the original artist.

Note that there are black-and-white versions of this available that scale down to smaller sizes and are better for letterheads or whatever you want to use it for: for the full range of logos take a look at Larry's web-page:

http://www.isc.tamu.edu/~lewing/linux/

Copyright (c) 2003-2011 QLogic Corporation
QLogic Linux qlge NIC Driver

You may modify and redistribute the device driver code under the GNU General Public License (a copy of which is attached hereto as Exhibit A) published by the Free Software Foundation (version 2).

EXHIBIT A

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.

Valid-License-Identifier: Linux-OpenIB
SPDX-URL: https://spdx.org/licenses/Linux-OpenIB.html
Usage-Guide:
To use the Linux Kernel Variant of OpenIB.org license put the following
SPDX tag/value pair into a comment according to the placement guidelines
in the licensing rules documentation:
SPDX-License-Identifier: Linux-OpenIB
License-Text:

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.

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 libuclient 2021-05-14

1.15.1 Available under license :

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

/*-
* Copyright (c) 1992, 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.
*
* 3. BSD Advertising Clause omitted per the July 22, 1999 licensing change
*   ftp://ftp.cs.berkeley.edu/pub/4bsd/README.Impt.License.Change
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/progress.c

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

Package: uclient-fetch
Version: 2021-05-14-6a6011df-1
Depends: libc, librt, libpthread, libuclient20201210
Provides: wget
Alternatives: 200:/usr/bin/wget:/bin/uclient-fetch
Source: package/libs/uclient
SourceName: uclient-fetch
License: ISC
Section: net
SourceDateEpoch: 1621028442
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 9284
Description: Tiny wget replacement using libuclient

Found in path(s):
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/ipkg-aarch64_cortex-a72/uclient-fetch/CONTROL/control
No license file was found, but licenses were detected in source scan.

Package: libuclient20201210
Version: 2021-05-14-6a6011df-1
Depends: libc, librt, libpthread, libubox20220515
Provides: libuclient
Source: package/libs/uclient
SourceName: libuclient
License: ISC
Section: libs
SourceDateEpoch: 1621028442
ABIVersion: 20201210
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 12104
Description: HTTP/1.1 client library

Found in path(s):
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/ipkg-aarch64_cortex-a72/libuclient/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* uclient - ustream based protocol client library
*
* Copyright (C) 2014 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/uclient-backend.h
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/uclient-utils.c
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/uclient-http.c
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/ipkg-install/usr/include/libubox/uclient.h
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/ipkg-install/usr/include/libubox/uclient-utils.h
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/uclient.h
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/uclient-fetch.c
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/uclient.c
* /opt/cola/permits/1601440482_1679668316.2135866/0/uclient-2021-05-14-6a6011df-zip/uclient-2021-05-14-6a6011df/uclient-utils.h

1.16 ubus 2022-06-01

1.16.1 Available under license :

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

/*
* Copyright (C) 2015 John Cripin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/libubus-acl.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011-2014 Felix Fietkau <nbd@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/libubus-internal.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/libubus.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/libubus-io.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/libubus.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_proto.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ipkg-install/usr/include/libubus.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/examples/server.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/libubus-req.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011 Felix Fietkau <nbd@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify

* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubus_common.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_event.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/examples/count.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_obj.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/cli.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_id.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ipkg-install/usr/include/ubus_common.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_id.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusmsg.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/examples/count.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ipkg-install/usr/include/ubusmsg.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_obj.h
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/examples/client.c

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

Package: libubus20220601
Version: 2022-06-01-2bebf93c-1
Depends: libc, librt, libpthread, libubox20220515
Provides: libubus
Source: package/system/ubus
SourceName: libubus
License: LGPL-2.1
Section: libs
SourceDateEpoch: 1654425492
ABIVersion: 20220601

Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 12313
Description: OpenWrt RPC client library

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ipkg-aarch64_cortex-a72/libubus/CONTROL/control
No license file was found, but licenses were detected in source scan.

Package: ubus
Version: 2022-06-01-2bebf93c-1
Depends: libc, librt, libpthread, libubus20220601, libblobmsg-json20220515, ubusd
Source: package/system/ubus
SourceName: ubus
License: LGPL-2.1
Section: base
SourceDateEpoch: 1654425492
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 7710
Description: OpenWrt RPC client utility

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ipkg-aarch64_cortex-a72/ubus/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 Jo-Philipp Wich <jow@openwrt.org>
* Copyright (C) 2012 John Crispin <blogic@openwrt.org>
* Copyright (C) 2016 Iain Fraser <iainf@netduma.com>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/lua/ubus.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_acl.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011-2012 Felix Fietkau <nbd@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/libubus-sub.c
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/libubus-obj.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 John Crispin <blogic@openwrt.org>
* Copyright (C) 2018 Hans Dedecker <dedeckeh@gmail.com>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_acl.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Felix Fietkau <nbd@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ubusd_monitor.c
No license file was found, but licenses were detected in source scan.

Package: ubusd
Version: 2022-06-01-2bebf93c-1
Depends: libc, librt, libpthread, libubox20220515, libblobmsg-json20220515
Source: package/system/ubus
SourceName: ubusd
License: LGPL-2.1
Section: base
Require-User: ubus=81:ubus=81
SourceDateEpoch: 1654425492
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 14363
Description: OpenWrt RPC daemon

Found in path(s):
* /opt/cola/permits/1601451248_1679668316.2312958/0/ubus-2022-06-01-2bebf93c-zip/ubus-2022-06-01-2bebf93c/ipkg-aarch64_cortex-a72/ubusd/CONTROL/control

1.17 libffi 3.4.2

1.17.1 Available under license :

The libffi source distribution contains certain code that is not part of libffi, and is only used as tooling to assist with the building and testing of libffi. This includes the msvcc.sh script used to wrap the Microsoft compiler with GNU compatible command-line options, make_sunver.pl, and the libffi test code distributed in the testsuite/libffi.bhaible directory. This code is distributed with libffi for the purpose of convenience only, and libffi is in no way derived from this code.

msvcc.sh an testsuite/libffi.bhaible are both distributed under the terms of the GNU GPL version 2, as below.

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.
libffi - Copyright (c) 1996-2021 Anthony Green, Red Hat, Inc and others.
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.

1.18 dnsmasq 2.86

1.18.1 Available under license :

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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) 19yy <name of author>

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

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

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

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

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

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

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

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

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

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

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

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

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>.

1.19 fstools 2022-06-02-93369be0-2

1.19.1 Available under license :

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

```
/*
* blkid.h - Interface for libblkid, a library to identify block devices
*
* Copyright (C) 2001 Andreas Dilger
* Copyright (C) 2003 Theodore Ts'o
* Copyright (C) 2008 Karel Zak <kzak@redhat.com>
*
* This library is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 2.1 of the License, or (at your option) any later version.
*
* This library is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public
* License along with this library; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
*/
```

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/blkid.h

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

```
/*
* No copyright is claimed.  This code is in the public domain; do with
* it what you wish.
*
* Written by Karel Zak <kzak@redhat.com>
*/
```

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/bitops.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/blkdev.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) International Business Machines Corp., 2006
*
* 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., 675 Mass Ave, Cambridge, MA 02139, USA.
*
* Author: Artem Bityutskiy
*
* UBI (Unsorted Block Images) library.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libubi/libubi.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libubi/libubi.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/ipkg-install/usr/include/libubi.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libubi/libubi_int.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007, 2008 Nokia Corporation
*
* 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., 675 Mass Ave, Cambridge, MA 02139, USA.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libubi/ubiutils-common.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008-2009 Karel Zak <kzak@redhat.com>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/superblocks.h
No license file was found, but licenses were detected in source scan.

/*
* encode.c - string conversion routines (mostly for compatibility with
*   udev/volume_id)
*
* Copyright (C) 2008 Kay Sievers <kay.sievers@vrfy.org>
* Copyright (C) 2009 Karel Zak <kzak@redhat.com>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/encode.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.org>
* Copyright (C) 2013 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify

* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/block.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/mkdev.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 Karel Zak <kzak@redhat.com>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/btrfs.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004-2008 Kay Sievers <kay.sievers@vrfy.org>
* Copyright (C) 2008 Karel Zak <kzak@redhat.com>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/hfs.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 Felix Fietkau <nbd@nbd.name>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*

* 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.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/rootdisk.c
No license file was found, but licenses were detected in source scan.

Package: fstools
Version: 2022-06-02-93369be0-2
Depends: libc, librt, libpthread, ubox, ubi-utils
Source: package/system/fstools
SourceName: fstools
License: GPL-2.0
Section: base
SourceDateEpoch: 1660337211
Maintainer: John Crispin <john@phrozen.org>
Architecture: aarch64_cortex-a72
Installed-Size: 22174
Description: OpenWrt filesystem tools

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/ipkg-aarch64_cortex-a72/fstools/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 1999, 2001 by Andries Brouwer
* Copyright (C) 1999, 2000, 2003 by Theodore Ts'o
* Copyright (C) 2008 Karel Zak <kzak@redhat.com>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/ext.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004 Kay Sievers <kay.sievers@vrfy.org>
* Copyright (C) 2008 Karel Zak <kzak@redhat.com>
*
* This file may be redistributed under the terms of the

* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/ntfs.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Daniel Golle <daniel@makrotopia.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/ubi.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 1999 by Andries Brouwer
* Copyright (C) 1999, 2000, 2003 by Theodore Ts'o
* Copyright (C) 2001 by Andreas Dilger
* Copyright (C) 2004 Kay Sievers <kay.sievers@vrfy.org>
* Copyright (C) 2008 Karel Zak <kzak@redhat.com>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/vfat.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/swap.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2008 Karel Zak <kzak@redhat.com>
*
* Inspired by libvolume_id by

*   Kay Sievers <kay.sievers@vrfy.org>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/jffs2.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/squashfs.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 Jo-Philipp Wich <jo@mein.io>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/probe.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/probe-libblkid.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/probe.h
No license file was found, but licenses were detected in source scan.

/*
* Low-level libblkid probing API
*
* Copyright (C) 2008-2009 Karel Zak <kzak@redhat.com>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/probe.c

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

/*
* Copyright (C) 2013 Alejandro Martinez Ruiz <alex@nowcomputing.com>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/f2fs.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2007 Nokia Corporation.
*
* This program is free software; you can redistribute it and/or modify it
* under the terms of the GNU General Public License version 2 as published by
* the Free Software Foundation.
*
* 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
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libubi/libubi-tiny.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright © International Business Machines Corp., 2006
*
* 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
*
* Author: Artem Bityutskiy ( )
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libubi/ubi-user.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/ipkg-install/usr/include/ubi-user.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 Rafa Miecki <rafal@milecki.pl>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/ubi.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/snapshot.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/ipkg-install/usr/include/libubi-tiny.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/overlay.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-

93369be0/libfstools/volume.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/find.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/ubi.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/jffs2reset.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/mtd.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/snapshot.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/mount_root.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/volume.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/libfstools.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/mount.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libubi/libubi-tiny.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/extroot.c
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libfstools/snapshot.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2009 Corentin Chary <corentincj@iksaif.net>
*
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/ubifs.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.org>
* Copyright (C) 2013 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/libblkid-tiny.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/ipkg-install/usr/include/libblkid-tiny.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (c) International Business Machines Corp., 2006
*
* 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
*
* Authors: Artem Bityutskiy ( )
*   Thomas Gleixner
*   Frank Haverkamp
*   Oliver Lohmann
*   Andreas Arnez
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libubi/ubi-media.h
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/ipkg-install/usr/include/ubi-media.h
No license file was found, but licenses were detected in source scan.

/*
* blkidP.h - Internal interfaces for libblkid
*
* Copyright (C) 2001 Andreas Dilger
* Copyright (C) 2003 Theodore Ts'o

*
* %Begin-Header%
* This file may be redistributed under the terms of the
* GNU Lesser General Public License.
* %End-Header%
*/

Found in path(s):
* /opt/cola/permits/1601366401_1679570261.4155128/0/fstools-2022-06-02-93369be0-zip/fstools-2022-06-02-93369be0/libblkid-tiny/blkidP.h

1.20 libnghttp2 1.44.0

1.20.1 Available under license :

Copyright (c) 2020 mruby developers

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

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

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

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

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

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
set yrange [0:]
set terminal pngcairo font 'Sans, 8' lw 1 size 1400,1024
set xtics rotate by -45
set style histogram errorbars gap 2 lw 1
set style fill solid border -1
If not otherwise noted, the extensions in this package are licensed
under the following license.

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

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

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

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

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

1.21 ubox 2021-08-03

1.21.1 Available under license :

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

/*
* Copyright (C) 2013 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/log/logd.c
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/log/syslog.h
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/log/syslog.c

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

/*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.org>
* Copyright (C) 2013 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/log/logread.c
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/kmodloader.c

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

/*
* Copyright (C) 2013 Jo-Philipp Wich <jow@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/validate/validate.c
No license file was found, but licenses were detected in source scan.

Package: getrandom
Version: 2021-08-03-205defb5-2
Depends: libc, librt, libpthread
Source: package/system/ubox
SourceName: getrandom
License: GPL-2.0
Section: base
SourceDateEpoch: 1643641405
Maintainer: John Crispin <john@phrozen.org>
Architecture: aarch64_cortex-a72
Installed-Size: 2644
Description: OpenWrt getrandom system helper

Found in path(s):
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/ipkg-aarch64_cortex-a72/getrandom/CONTROL/control
No license file was found, but licenses were detected in source scan.

Package: logd
Version: 2021-08-03-205defb5-2
Depends: libc, librt, libpthread, libubox20220515, libubus20220601, libblobmsg-json20220515
Source: package/system/ubox
SourceName: logd
License: GPL-2.0
Section: base
Require-User: logd=514:logd=514
SourceDateEpoch: 1643641405
Maintainer: John Crispin <john@phrozen.org>

Architecture: aarch64_cortex-a72
Installed-Size: 13382
Description: OpenWrt system log implementation

Found in path(s):
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/ipkg-aarch64_cortex-a72/logd/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
*   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; version 2 of the License
*
*   This program is distributed in the hope that it will be useful,
*   but WITHOUT ANY WARRANTY; without even the implied warranty of
*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See 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.
*
*   Copyright (C) 2012 John Crispin <blogic@openwrt.org>
*/

Found in path(s):
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/lsbloader.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2016 Etienne Champetier <champetier.etienne@gmail.com>
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/getrandom.c

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

Package: ubox
Version: 2021-08-03-205defb5-2
Depends: libc, librt, libpthread, libubox20220515, ubusd, ubus, libubus20220601, libuci20130104
Alternatives: 100:/sbin/rmmod:/sbin/kmodloader, 100:/sbin/insmod:/sbin/kmodloader, 100:/sbin/lsmod:/sbin/kmodloader, 100:/sbin/modinfo:/sbin/kmodloader, 100:/sbin/modprobe:/sbin/kmodloader
Source: package/system/ubox
SourceName: ubox
License: GPL-2.0
Section: base
SourceDateEpoch: 1643641405
Maintainer: John Crispin <john@phrozen.org>
Architecture: aarch64_cortex-a72
Installed-Size: 19388
Description: OpenWrt system helper toolbox

Found in path(s):
* /opt/cola/permits/1601450456_1679570361.7577162/0/ubox-2021-08-03-205defb5-zip/ubox-2021-08-03-205defb5/ipkg-aarch64_cortex-a72/ubox/CONTROL/control

1.22 ncurses 6.3

1.22.1 Available under license :

Upstream source https://invisible-island.net/ncurses/ncurses.html
This package is used for testing builds of ncurses.

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2020,2021 by Thomas E. Dickey
Copyright: 1998-2016,2017 Free Software Foundation, Inc.
Licence: X11

Files: aclocal.m4 package
Copyright: 1996-2020,2021 by Thomas E. Dickey
Licence: X11

Files: doc/html/NCURSES-Programming-HOWTO.html
Copyright: 2001 by Pradeep Padala
Licence: X11

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, distribute with modifications, 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 ABOVE 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(s) of the above copyright
holders shall not be used in advertising or otherwise to promote the
sale, use or other dealings in this Software without prior written
authorization.

Files: install-sh
Copyright: 1994 X Consortium
Licence: X11

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
X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNEC-
TION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not
be used in advertising or otherwise to promote the sale, use or other deal-
ings in this Software without prior written authorization from the X Consor-
tium.

Files: progs/tset.c ncurses/tinfo/read_termcap.c
Copyright: 1980,1991,1992,1993 The Regents of the University of California.
License: BSD

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 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.

-- vile: txtmode file-encoding=utf-8
Copyright 2018-2020,2021 Thomas E. Dickey
Copyright 1998-2017,2018 Free Software Foundation, 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, distribute with modifications, 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 ABOVE 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(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

-- vile:txtmode fc=72
-- $Id: COPYING,v 1.10 2021/01/01 09:54:30 tom Exp $
Upstream source https://invisible-island.net/ncurses/Ada95.html

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2020,2021 by Thomas E. Dickey
Copyright: 1998-2016,2017 Free Software Foundation, Inc.
Licence: X11

Files: aclocal.m4 package
Copyright: 2010-2020,2021 by Thomas E. Dickey
Licence: X11

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, distribute with modifications, 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 ABOVE 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(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh
Copyright: 1994 X Consortium
Licence: X11

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 X CONSORTIUM 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 the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent `make' implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8
Upstream source https://invisible-island.net/ncurses/ncurses-examples.html

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2020,2021 Thomas E. Dickey

Copyright: 1998-2016,2017 Free Software Foundation, Inc.
Licence: X11

Files: aclocal.m4 package
Copyright: 2003-2019,2020 by Thomas E. Dickey
Licence: X11

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, distribute with modifications, 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 ABOVE 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(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh
Copyright: 1994 X Consortium
Licence: X11

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 X CONSORTIUM 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 the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent `make' implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written from scratch. It can only install one file at a time, a restriction shared with many OS's install programs.

On Debian systems, the complete text of the GNU General Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8
Upstream source https://invisible-island.net/ncurses/ncurses-examples.html

Current ncurses maintainer: Thomas Dickey <dickey@invisible-island.net>

Files: *
Copyright: 2017-2020,2021 Thomas E. Dickey
Copyright: 1998-2016,2017 Free Software Foundation, Inc.
Licence: X11

Files: aclocal.m4 package
Copyright: 2003-2020,2021 by Thomas E. Dickey
Licence: X11

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, distribute with modifications, 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 ABOVE 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(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Files: install-sh
Copyright: 1994 X Consortium
Licence: X11

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 X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNEC-TION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other deal-ings in this Software without prior written authorization from the X Consor-tium.

FSF changes to this file are in the public domain.

Calling this script install-sh is preferred over install.sh, to prevent `make' implicit rules from creating a file called install from it when there is no Makefile.

This script is compatible with the BSD install script, but was written

from scratch. It can only install one file at a time, a restriction
shared with many OS's install programs.

On Debian systems, the complete text of the GNU General
Public License can be found in '/usr/share/common-licenses/GPL-2'

-- vile: txtmode file-encoding=utf-8

1.23 procd 2021-03-08-2cfc26f8-1

1.23.1 Available under license :

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

/*
* Copyright (C) 2015 Etienne CHAMPETIER <champetier.etienne@gmail.com>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/capabilities.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2004-2006 Kay Sievers <kay@vrfy.org>
* Copyright (C) 2006 Hannes Reinecke <hare@suse.de>
*
* 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 version 2 of the License.
*
* This program is distributed in the hope that it will be useful, but
* WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See 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.

*
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/plug/udevtrigger.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2019 Paul Spooren <mail@aparcar.de>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/container.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 John Crispin <blogic@openwrt.org>
* Copyright (C) 2015 Etienne Champetier <champetier.etienne@gmail.com>
* Copyright (C) 2020 Daniel Golle <daniel@makrotopia.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/fs.c
No license file was found, but licenses were detected in source scan.

/*
* runqueue-example.c

*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/rcS.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2017 Matthias Schiffer <mschiffer@universe-factory.net>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/sysupgrade.h
No license file was found, but licenses were detected in source scan.

Package: procd-ujail
Version: 2021-03-08-2cfc26f8-1
Depends: libc, librt, libpthread, libubox20220515, libubus20220601, libblobmsg-json20220515
Source: package/system/procd
SourceName: procd-ujail
License: GPL-2.0
Section: base
SourceDateEpoch: 1654425472
Maintainer: John Crispin <john@phrozen.org>

Architecture: aarch64_cortex-a72
Installed-Size: 32039
Description: OpenWrt process jail helper

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/ipkg-aarch64_cortex-a72/procd-ujail/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.org>
* Copyright (C) 2013 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/watchdog.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/initd/preinit.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/service/trigger.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/service/service.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/signal.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/state.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/system.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/service/instance.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/plug/hotplug.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/utils/utils.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/initd/early.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/utils/askfirst.c

* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/service/instance.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/upgraded/upgraded.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/rcS.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/initd/mkdev.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/plug/coldplug.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/watchdog.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/plug/hotplug.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/utils/utils.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/procd.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/log.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/service/service.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/ubus.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/inittab.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/service/watch.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/procd.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/initd/init.c

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

/*
* Copyright (C) 2015 John Crispin <blogic@openwrt.org>
* Copyright (C) 2020 Daniel Golle <daniel@makrotopia.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/jail.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Etienne Champetier <champetier.etienne@gmail.com>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/fs.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/jail.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/elf.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/preload.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/trace/trace.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/elf.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-

2cfc26f8/jail/seccomp.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/trace/preload.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/seccomp-syscalls-helpers.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/preload.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/log.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Daniel Golle <daniel@makrotopia.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*
* reads unified cgroup config as proposed in
* https://github.com/opencontainers/runtime-spec/pull/1040
* attempt conversion from cgroup1 -> cgroup2
* https://github.com/containers/crun/blob/0.14.1/crun.1.md#cgroup-v2
*
* ToDo:
* - convert cgroup1 devices to eBPF program
* - convert cgroup1 net_prio and net_cls to eBPF program
* - rdma (anyone?) intelrdt (anyone?)
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/cgroups.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2015 Etienne CHAMPETIER <champetier.etienne@gmail.com>
* Copyright (C) 2020 Daniel Golle <daniel@makrotopia.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/capabilities.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2021 Daniel Golle <daniel@makrotopia.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/hotplug-dispatch.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.org>
* Copyright (C) 2013 John Crispin <blogic@openwrt.org>
* Copyright (C) 2017 Matthias Schiffer <mschiffer@universe-factory.net>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/sysupgrade.c
No license file was found, but licenses were detected in source scan.

Package: procd
Version: 2021-03-08-2cfc26f8-1
Depends: libc, librt, libpthread, ubusd, ubus, libjson-script20220515, ubox, libubox20220515, libubus20220601, libblobmsg-json20220515, libjson-c5, jshn
Conflicts: procd-selinux
Source: package/system/procd
SourceName: procd
License: GPL-2.0
Section: base
Require-User: :dialout=20 :audio=29
SourceDateEpoch: 1654425472
Maintainer: John Crispin <john@phrozen.org>
Architecture: aarch64_cortex-a72
Installed-Size: 61156
Description: OpenWrt system process manager

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/ipkg-aarch64_cortex-a72/procd/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 John Crispin <blogic@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/initd/init.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/service/validate.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2020 Daniel Golle <daniel@makrotopia.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU Lesser General Public License version 2.1
* as published by the Free Software Foundation
*

* 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.
*/

Found in path(s):
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/uxc.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/console.c
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/seccomp-oci.h
* /opt/cola/permits/1601395261_1679570349.9239135/0/procd-2021-03-08-2cfc26f8-zip/procd-2021-03-08-2cfc26f8/jail/cgroups.h

1.24 tpm2-tss-tcti 3.1.0

1.24.1 Available under license :

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.

1.25 curl 7.85.0

1.25.1 Available under license :

1.26 glib 2.70.5

1.26.1 Available under license :

PURPOSE. SHOULD THE WORK PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

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 WORK 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 WORK, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

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

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

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

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

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

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

a) The modified work must itself be a software library.

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

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

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

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

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

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which

must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

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

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the

Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

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

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

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

11. If, as a consequence of a court judgment or allegation of patent

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

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

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

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

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

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

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

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

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

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

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.27 d-bus 1.13.18

1.27.1 Available under license :

D-Bus is licensed to you under your choice of the Academic Free
License version 2.1, or the GNU General Public License version 2
(or, at your option any later version).

Both licenses are included here. Some of the standalone binaries are
under the GPL only; in particular, but not limited to,
tools/dbus-cleanup-sockets.c and test/decode-gcov.c. Each source code
file is marked with the proper copyright information - if you find a
file that isn't marked please bring it to our attention.

The Academic Free License

v. 2.1

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Academic Free License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

a) to reproduce the Original Work in copies;

b) to prepare derivative works ("Derivative Works") based upon the Original Work;

c) to distribute copies of the Original Work and Derivative Works to the public;

d) to perform the Original Work publicly; and

e) to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repository in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated

herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) This section intentionally omitted.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately proceeding sentence, the Original Work is provided under this License on an "AS IS" BASIS and WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor'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.

9) Acceptance and Termination. If You distribute copies of the

Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) 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.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, 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 that controls, is controlled by, or is under common control with you. For 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.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by You.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

--
END OF ACADEMIC FREE LICENSE. The following is intended to describe the essential differences between the Academic Free License (AFL) version 1.0 and other open source licenses:

The Academic Free License is similar to the BSD, MIT, UoI/NCSA and Apache licenses in many respects but it is intended to solve a few problems with those licenses.

* The AFL is written so as to make it clear what software is being licensed (by the inclusion of a statement following the copyright notice in the software). This way, the license functions better than a template license. The BSD, MIT and UoI/NCSA licenses apply to unidentified software.

* The AFL contains a complete copyright grant to the software. The BSD and Apache licenses are vague and incomplete in that respect.

* The AFL contains a complete patent grant to the software. The BSD, MIT, UoI/NCSA and Apache licenses rely on an implied patent license and contain no explicit patent grant.

* The AFL makes it clear that no trademark rights are granted to the licensor's trademarks. The Apache license contains such a provision, but the BSD, MIT and UoI/NCSA licenses do not.

* The AFL includes the warranty by the licensor that it either owns the copyright or that it is distributing the software under a license. None of the other licenses contain that warranty. All other warranties are disclaimed,

as is the case for the other licenses.

* The AFL is itself copyrighted (with the right granted to copy and distribute without modification). This ensures that the owner of the copyright to the license will control changes. The Apache license contains a copyright notice, but the BSD, MIT and UoI/NCSA licenses do not.

--

START OF GNU GENERAL PUBLIC LICENSE

--

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

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 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 Library General
Public License instead of this License.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the 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.

1.28 i2c-tools 4.3

1.28.1 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.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

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

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

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

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

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

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

a) The modified work must itself be a software library.

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

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

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

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

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

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

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

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and

therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

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

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

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

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

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

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

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

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

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

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

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

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

Lesser General Public License for more details.

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

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.29 libarchive 3.6.1

1.29.1 Available under license :

The libarchive distribution as a whole is Copyright by Tim Kientzle
and is subject to the copyright notice reproduced at the bottom of
this file.

Each individual file in this distribution should have a clear
copyright/licensing statement at the beginning of the file. If any do
not, please let me know and I will rectify it. The following is
intended to summarize the copyright status of the individual files;
the actual statements in the files are controlling.

* Except as listed below, all C sources (including .c and .h files)
 and documentation files are subject to the copyright notice reproduced
 at the bottom of this file.

* The following source files are also subject in whole or in part to
 a 3-clause UC Regents copyright; please read the individual source
 files for details:
  libarchive/archive_read_support_filter_compress.c
  libarchive/archive_write_add_filter_compress.c
  libarchive/mtree.5

* The following source files are in the public domain:
  libarchive/archive_getdate.c

* The following source files are triple-licensed with the ability to choose
 from CC0 1.0 Universal, OpenSSL or Apache 2.0 licenses:
  libarchive/archive_blake2.h
  libarchive/archive_blake2_impl.h
  libarchive/archive_blake2s_ref.c
  libarchive/archive_blake2sp_ref.c

* The build files---including Makefiles, configure scripts,
 and auxiliary scripts used as part of the compile process---have
 widely varying licensing terms. Please check individual files before
 distributing them to see if those restrictions apply to you.

I intend for all new source code to use the license below and hope over
time to replace code with other licenses with new implementations that
do use the license below. The varying licensing of the build scripts
seems to be an unavoidable mess.

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
  in this position and unchanged.
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(S) ``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(S) 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.

1.30 netifd 2022-08-25

1.30.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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.
/*
* netifd - network interface daemon
* Copyright (C) 2012 Felix Fietkau <nbd@openwrt.org>
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License version 2
* as published by the Free Software Foundation
*
* 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.
*/

1.31 jsonfilter 2018-02-04

1.31.1 Available under license :

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

Package: jsonfilter
Version: 2018-02-04-c7e938d6-1
Depends: libc, librt, libpthread, libubox20220515, libjson-c5
Source: package/utils/jsonfilter
SourceName: jsonfilter
License: ISC
Section: base
SourceDateEpoch: 1645878990
Maintainer: Jo-Philipp Wich <jo@mein.io>
Architecture: aarch64_cortex-a72
Installed-Size: 12064
Description: OpenWrt JSON filter utility

Found in path(s):
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/ipkg-aarch64_cortex-a72/jsonfilter/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013-2014 Jo-Philipp Wich <jo@mein.io>
*
* 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.
*/

Found in path(s):
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/matcher.c
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/main.c
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/ast.c
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/lexer.c
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/lexer.h
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/matcher.h
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/parser.y
* /opt/cola/permits/1601420371_1679668607.7526705/0/jsonfilter-2018-02-04-c7e938d6-zip/jsonfilter-2018-02-04-c7e938d6/ast.h

1.32 pcre 8.45

1.32.1 Available under license :

PCRE LICENCE

Please see the file LICENCE in the PCRE distribution for licensing details.

End
PCRE LICENCE

PCRE is a library of functions to support regular expressions whose syntax
and semantics are as close as possible to those of the Perl 5 language.

Release 8 of PCRE is distributed under the terms of the "BSD" licence, as
specified below. The documentation for PCRE, supplied in the "doc"
directory, is distributed under the same terms as the software itself. The data
in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also included in the distribution is a set of C++ wrapper functions, and a just-in-time compiler that can be used to optimize pattern matching. These are both optional features that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS

Written by: Philip Hazel
Email local part: Philip.Hazel
Email domain: gmail.com

University of Cambridge Computing Service,
Cambridge, England.

PCRE JUST-IN-TIME COMPILATION SUPPORT

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

STACK-LESS JUST-IN-TIME COMPILER

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

THE C++ WRAPPER FUNCTIONS

Contributed by: Google Inc.

THE "BSD" LICENCE

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 University of Cambridge nor the name of Google Inc. nor the names of their 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.

End

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

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the 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.

1.33 mtd-utils 2.1.4

1.33.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

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

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

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

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

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

and give any other recipients of the Program a copy of this License along with the Program.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NO WARRANTY

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

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY

YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

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

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

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

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

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

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

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

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

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

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

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

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

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

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
Copyright (c) 2000-2007 by Nicolas Devillard.
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.

1.34 ntfs-3g 2021.8.22

1.34.1 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 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 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 Library General
Public License instead of this License.

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

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 Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, 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 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 a program 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.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, 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 companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

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, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library 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 compile 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) 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.

c) 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.

d) 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 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.

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 to 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 Library 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

Appendix: 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 Library 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 Library General Public License for more details.

You should have received a copy of the GNU Library 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

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

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

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.35 readline 8.1

1.35.1 Available under license :

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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., 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.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

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>.
@c The GNU Free Documentation License.
@center Version 1.3, 3 November 2008

@c This file is intended to be included within another document,
@c hence no sectioning command or @node.

@display
Copyright @copyright{} 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
@uref{http://fsf.org/}

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

@enumerate 0
@item
PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document @dfn{free} in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ``copyleft'', which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

@item
APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ``Document'', below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ``you''. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ``Modified Version'' of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ``Secondary Section'' is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ``Invariant Sections'' are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ``Cover Texts'' are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ``Transparent'' copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available

drawing editor, and that is suitable for input to text formatters or
for automatic translation to a variety of formats suitable for input
to text formatters. A copy made in an otherwise Transparent file
format whose markup, or absence of markup, has been arranged to thwart
or discourage subsequent modification by readers is not Transparent.
An image format is not Transparent if used for any substantial amount
of text. A copy that is not ``Transparent'' is called ``Opaque''.

Examples of suitable formats for Transparent copies include plain
@sc{ascii} without markup, Texinfo input format, La@TeX{} input
format, @acronym{SGML} or @acronym{XML} using a publicly available
@acronym{DTD}, and standard-conforming simple @acronym{HTML},
PostScript or @acronym{PDF} designed for human modification. Examples
of transparent image formats include @acronym{PNG}, @acronym{XCF} and
@acronym{JPG}. Opaque formats include proprietary formats that can be
read and edited only by proprietary word processors, @acronym{SGML} or
@acronym{XML} for which the @acronym{DTD} and/or processing tools are
not generally available, and the machine-generated @acronym{HTML},
PostScript or @acronym{PDF} produced by some word processors for
output purposes only.

The ``Title Page'' means, for a printed book, the title page itself,
plus such following pages as are needed to hold, legibly, the material
this License requires to appear in the title page. For works in
formats which do not have any title page as such, ``Title Page'' means
the text near the most prominent appearance of the work's title,
preceding the beginning of the body of the text.

The ``publisher'' means any person or entity that distributes copies
of the Document to the public.

A section ``Entitled XYZ'' means a named subunit of the Document whose
title either is precisely XYZ or contains XYZ in parentheses following
text that translates XYZ in another language. (Here XYZ stands for a
specific section name mentioned below, such as ``Acknowledgements'',
``Dedications'', ``Endorsements'', or ``History''.) To ``Preserve the Title''
of such a section when you modify the Document means that it remains a
section ``Entitled XYZ'' according to this definition.

The Document may include Warranty Disclaimers next to the notice which
states that this License applies to the Document. These Warranty
Disclaimers are considered to be included by reference in this
License, but only as regards disclaiming warranties: any other
implication that these Warranty Disclaimers may have is void and has
no effect on the meaning of this License.

@item
VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

@item
COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

@item
MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

@enumerate A
@item
Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

@item
List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

@item
State on the Title page the name of the publisher of the Modified Version, as the publisher.

@item
Preserve all the copyright notices of the Document.

@item
Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

@item
Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

@item
Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

@item
Include an unaltered copy of this License.

@item
Preserve the section Entitled ``History'', Preserve its Title, and add
to it an item stating at least the title, year, new authors, and
publisher of the Modified Version as given on the Title Page. If
there is no section Entitled ``History'' in the Document, create one
stating the title, year, authors, and publisher of the Document as
given on its Title Page, then add an item describing the Modified
Version as stated in the previous sentence.

@item
Preserve the network location, if any, given in the Document for
public access to a Transparent copy of the Document, and likewise
the network locations given in the Document for previous versions
it was based on. These may be placed in the ``History'' section.
You may omit a network location for a work that was published at
least four years before the Document itself, or if the original
publisher of the version it refers to gives permission.

@item
For any section Entitled ``Acknowledgements'' or ``Dedications'', Preserve
the Title of the section, and preserve in the section all the
substance and tone of each of the contributor acknowledgements and/or
dedications given therein.

@item
Preserve all the Invariant Sections of the Document,
unaltered in their text and in their titles. Section numbers
or the equivalent are not considered part of the section titles.

@item
Delete any section Entitled ``Endorsements''. Such a section
may not be included in the Modified Version.

@item
Do not retitle any existing section to be Entitled ``Endorsements'' or
to conflict in title with any Invariant Section.

@item
Preserve any Warranty Disclaimers.
@end enumerate

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document, you may at your option designate some or all

of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ``Endorsements'', provided it contains nothing but endorsements of your Modified Version by various parties---for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

@item
COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ``History'' in the various original documents, forming one section Entitled ``History''; likewise combine any sections Entitled ``Acknowledgements'', and any sections Entitled ``Dedications''. You must delete all sections Entitled ``Endorsements.''

@item
COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

@item
AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ``aggregate'' if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

@item
TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between

the translation and the original version of this License or a notice
or disclaimer, the original version will prevail.

If a section in the Document is Entitled ``Acknowledgements'',
``Dedications'', or ``History'', the requirement (section 4) to Preserve
its Title (section 1) will typically require changing the actual
title.

@item
TERMINATION

You may not copy, modify, sublicense, or distribute the Document
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense, or distribute it is void, and
will automatically terminate your rights under this License.

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, receipt of a copy of some or all of the same material does
not give you any rights to use it.

@item
FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions
of the GNU Free Documentation 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. See
@uref{http://www.gnu.org/copyleft/}.

Each version of the License is given a distinguishing version number.
If the Document specifies that a particular numbered version of this
License ``or any later version'' applies to it, you have the option of

following the terms and conditions either of that specified version or
of any later version that has been published (not as a draft) by the
Free Software Foundation. If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation. If the Document
specifies that a proxy can decide which future versions of this
License can be used, that proxy's public statement of acceptance of a
version permanently authorizes you to choose that version for the
Document.

@item
RELICENSING

``Massive Multiauthor Collaboration Site'' (or ``MMC Site'') means any
World Wide Web server that publishes copyrightable works and also
provides prominent facilities for anybody to edit those works. A
public wiki that anybody can edit is an example of such a server. A
``Massive Multiauthor Collaboration'' (or ``MMC'') contained in the
site means any set of copyrightable works thus published on the MMC
site.

``CC-BY-SA'' means the Creative Commons Attribution-Share Alike 3.0
license published by Creative Commons Corporation, a not-for-profit
corporation with a principal place of business in San Francisco,
California, as well as future copyleft versions of that license
published by that same organization.

``Incorporate'' means to publish or republish a Document, in whole or
in part, as part of another Document.

An MMC is ``eligible for relicensing'' if it is licensed under this
License, and if all works that were first published under this License
somewhere other than this MMC, and subsequently incorporated in whole
or in part into the MMC, (1) had no cover texts or invariant sections,
and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site
under CC-BY-SA on the same site at any time before August 1, 2009,
provided the MMC is eligible for relicensing.

@end enumerate

@page
@heading ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of
the License in the document and put the following copyright and
license notices just after the title page:

```
@smallexample
@group
 Copyright (C)  @var{year}  @var{your name}.
 Permission is granted to copy, distribute and/or modify this document
 under the terms of the GNU Free Documentation License, Version 1.3
 or any later version published by the Free Software Foundation;
 with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
 Texts.  A copy of the license is included in the section entitled ``GNU
 Free Documentation License''.
@end group
@end smallexample

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts,
replace the ``with@dots{}Texts.'' line with this:

@smallexample
@group
   with the Invariant Sections being @var{list their titles}, with
   the Front-Cover Texts being @var{list}, and with the Back-Cover Texts
   being @var{list}.
@end group
@end smallexample

If you have Invariant Sections without Cover Texts, or some other
combination of the three, merge those two alternatives to suit the
situation.

If your document contains nontrivial examples of program code, we
recommend releasing these examples in parallel under your choice of
free software license, such as the GNU General Public License,
to permit their use in free software.

@c Local Variables:
@c ispell-local-pdict: "ispell-dict"
@c End:
```

1.36 zlib 1.2.11

1.36.1 Available under license :

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.

1.37 libubox 2022-05-15

1.37.1 Available under license :

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

/*
* runqueue.c - a simple task queueing/completion tracking helper
*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/runqueue.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/runqueue.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/runqueue.h

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

/*-
* Copyright (c) 2011 Felix Fietkau <nbd@openwrt.org>
* Copyright (c) 2010 Isilon Systems, Inc.
* Copyright (c) 2010 iX Systems, Inc.
* Copyright (c) 2010 Panasas, 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 unmodified, 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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/list.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/list.h
No license file was found, but licenses were detected in source scan.

/*
* uloop - event loop implementation
*
* Copyright (C) 2010-2016 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/uloop.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/uloop-kqueue.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/uloop-epoll.c
No license file was found, but licenses were detected in source scan.

/*
* ulog - simple logging functions
*
* Copyright (C) 2015 Jo-Philipp Wich <jow@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ulog.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ulog.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/ulog.h
No license file was found, but licenses were detected in source scan.

Copyright 2008 Kate Ward. All Rights Reserved.
Released under the LGPL (GNU Lesser General Public License)

Found in path(s):

* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/examples/shunit2
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/json_script.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/json_script.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/json_script.c
No license file was found, but licenses were detected in source scan.

/*
* safe_list - linked list protected against recursive iteration with deletes
*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):

* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/safe_list.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/safe_list.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/safe_list.h
No license file was found, but licenses were detected in source scan.

/*
* utils - misc libubox utility functions
*
* Copyright (C) 2012 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/utils.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/utils.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/utils.h
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2011-2013 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/jshn.c
No license file was found, but licenses were detected in source scan.

/*
* base64 - libubox base64 functions
*
* Copyright (C) 2015 Felix Fietkau <nbd@openwrt.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.
*/
/*
* Copyright (c) 1996 by Internet Software Consortium.
*
* 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 INTERNET SOFTWARE CONSORTIUM DISCLAIMS
* ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE
* CONSORTIUM 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.
*/
/*
* Portions Copyright (c) 1995 by International Business Machines, Inc.
*
* International Business Machines, Inc. (hereinafter called IBM) grants
* permission under its copyrights to use, copy, modify, and distribute this
* Software with or without fee, provided that the above copyright notice and
* all paragraphs of this notice appear in all copies, and that the name of IBM

* not be used in connection with the marketing of any product incorporating
* the Software or modifications thereof, without specific, written prior
* permission.
*
* To the extent it has a right to do so, IBM grants an immunity from suit
* under its patents, if any, for the use, sale or manufacture of products to
* the extent that such products are used for performing Domain Name System
* dynamic updates in TCP/IP networks by means of the Software. No immunity is
* granted for any product per se or for any other function of any product.
*
* THE SOFTWARE IS PROVIDED "AS IS", AND IBM DISCLAIMS ALL WARRANTIES,
* INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
* PARTICULAR PURPOSE. IN NO EVENT SHALL IBM BE LIABLE FOR ANY SPECIAL,
* DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER ARISING
* OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE, EVEN
* IF IBM IS APPRISED OF THE POSSIBILITY OF SUCH DAMAGES.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/base64.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/avl-cmp.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/vlist.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/avl-cmp.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/vlist.h

* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/vlist.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/avl-cmp.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2010-2012 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/blobmsg.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/blobmsg_json.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/blobmsg.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/blobmsg_json.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/blobmsg_json.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/blobmsg.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2012 John Crispin <blogic@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/lua/uloop.c
No license file was found, but licenses were detected in source scan.

Package: libblobmsg-json20220515
Version: 2022-05-15-d2223ef9-1
Depends: libc, librt, libpthread, libjson-c5, libubox20220515
Provides: libblobmsg-json
Source: package/libs/libubox
SourceName: libblobmsg-json
License: ISC
Section: libs
SourceDateEpoch: 1654630618
ABIVersion: 20220515
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 4795
Description: blobmsg <-> json conversion library

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-aarch64_cortex-a72/libblobmsg-json/CONTROL/control
No license file was found, but licenses were detected in source scan.

/*
* ustream - library for stream buffer management
*
* Copyright (C) 2012 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/ustream.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ustream.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ustream-fd.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ustream.h
No license file was found, but licenses were detected in source scan.

Package: jshn
Version: 2022-05-15-d2223ef9-1
Depends: libc, librt, libpthread, libjson-c5, libubox20220515, libblobmsg-json20220515
Source: package/libs/libubox
SourceName: jshn
License: ISC
Section: utils
SourceDateEpoch: 1654630618
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 7890
Description: Library for parsing and generating JSON from shell scripts

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-aarch64_cortex-a72/jshn/CONTROL/control
No license file was found, but licenses were detected in source scan.

Found in path(s):

* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/kvlist.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/kvlist.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/kvlist.h

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

/*
* Copyright (C) 2014 Felix Fietkau <nbd@openwrt.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.
*/
/*
* This is an OpenSSL-compatible implementation of the RSA Data Security, Inc.
* MD5 Message-Digest Algorithm (RFC 1321).
*
* Homepage:
* http://openwall.info/wiki/people/solar/software/public-domain-source-code/md5
*
* Author:
* Alexander Peslyak, better known as Solar Designer <solar at openwall.com>
*
* This software was written by Alexander Peslyak in 2001. No copyright is
* claimed, and the software is hereby placed in the public domain.
* In case this attempt to disclaim copyright and place the software in the
* public domain is deemed null and void, then the software is
* Copyright (c) 2001 Alexander Peslyak and it is hereby released to the
* general public under the following terms:
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted.
*
* There's ABSOLUTELY NO WARRANTY, express or implied.
*
* (This is a heavily cut-down "BSD license".)
*
* This differs from Colin Plumb's older public domain implementation in that

* no exactly 32-bit integer data type is required (any 32-bit or wider
* unsigned integer data type will do), there's no compile-time endianness
* configuration, and the function prototypes match OpenSSL's. No code from
* Colin Plumb's implementation has been reused; this comment merely compares
* the properties of the two independent implementations.
*
* The primary goals of this implementation are portability and ease of use.
* It is meant to be fast, but not as fast as possible. Some known
* optimizations are not included to reduce source code size and avoid
* compile-time configuration.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/md5.c
No license file was found, but licenses were detected in source scan.

/*
* uloop - event loop implementation
*
* Copyright (C) 2010-2013 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/uloop.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/uloop.h
No license file was found, but licenses were detected in source scan.

/*
* runqueue-example.c
*
* Copyright (C) 2013 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/tests/test-runqueue.c
No license file was found, but licenses were detected in source scan.

/*
* Copyright (C) 2014 Felix Fietkau <nbd@openwrt.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.
*/
/*
* This is an OpenSSL-compatible implementation of the RSA Data Security, Inc.
* MD5 Message-Digest Algorithm (RFC 1321).
*
* Homepage:
* http://openwall.info/wiki/people/solar/software/public-domain-source-code/md5
*
* Author:
* Alexander Peslyak, better known as Solar Designer <solar at openwall.com>
*
* This software was written by Alexander Peslyak in 2001. No copyright is
* claimed, and the software is hereby placed in the public domain.
* In case this attempt to disclaim copyright and place the software in the
* public domain is deemed null and void, then the software is
* Copyright (c) 2001 Alexander Peslyak and it is hereby released to the
* general public under the following terms:

*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted.
*
* There's ABSOLUTELY NO WARRANTY, express or implied.
*
* See md5.c for more information.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/md5.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/md5.h
No license file was found, but licenses were detected in source scan.

Package: libubox20220515
Version: 2022-05-15-d2223ef9-1
Depends: libc, librt, libpthread
Provides: libubox
Source: package/libs/libubox
SourceName: libubox
License: ISC
Section: libs
SourceDateEpoch: 1654630618
ABIVersion: 20220515
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72
Installed-Size: 25465
Description: Basic utility library

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-aarch64_cortex-a72/libubox/CONTROL/control
No license file was found, but licenses were detected in source scan.

Package: libjson-script20220515
Version: 2022-05-15-d2223ef9-1
Depends: libc, librt, libpthread, libubox20220515
Provides: libjson-script
Source: package/libs/libubox
SourceName: libjson-script
License: ISC
Section: utils
SourceDateEpoch: 1654630618
ABIVersion: 20220515
Maintainer: Felix Fietkau <nbd@nbd.name>
Architecture: aarch64_cortex-a72

Installed-Size: 6329
Description: Minimalistic JSON based scripting engine

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-aarch64_cortex-a72/libjson-script/CONTROL/control
No license file was found, but licenses were detected in source scan.

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/avl.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/avl.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/avl.h
No license file was found, but licenses were detected in source scan.

/*
* usock - socket helper functions
*
* Copyright (C) 2010 Steven Barth <steven@midlink.org>
* Copyright (C) 2011-2012 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/usock.c
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/usock.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/usock.h
No license file was found, but licenses were detected in source scan.

/*
* blob - library for generating/parsing tagged binary data
*
* Copyright (C) 2010 Felix Fietkau <nbd@openwrt.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.
*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/blob.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/ipkg-install/usr/include/libubox/blob.h
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/blob.c
No license file was found, but licenses were detected in source scan.

```
# $Id: shunit2 189 2008-07-11 11:46:54Z kate.ward@forestent.com $
# vim:et:ft=sh:sts=2:sw=2
# vim:foldmethod=marker:foldmarker=/**,*/
#
#/**
# <?xml version="1.0" encoding="UTF-8"?>
# <s:shelldoc xmlns:s="http://www.forestent.com/projects/shelldoc/xsl/2005.0">
# <s:header>
# shUnit 2.1.4
# Shell Unit Test Framework
#
# http://shunit2.sourceforge.net/
#
# written by Kate Ward &lt;kate.ward@forestent.com&gt;
# released under the LGPL
#
# this module implements a xUnit based unit test framework similar to JUnit
# </s:header>
#*/

SHUNIT_VERSION='2.1.4'

_shunit_warn() { echo "shunit2:WARN $@" >&2; }
_shunit_error() { echo "shunit2:ERROR $@" >&2; }
_shunit_fatal() { echo "shunit2:FATAL $@" >&2; }

SHUNIT_TRUE=0
SHUNIT_FALSE=1
SHUNIT_ERROR=2

# specific shell checks
if [ -n "${ZSH_VERSION:-}" ]; then
```

```
 setopt |grep "^shwordsplit$" >/dev/null
 if [ $? -ne ${SHUNIT_TRUE} ]; then
   _shunit_fatal 'zsh shwordsplit option is required for proper operation'
   exit ${SHUNIT_ERROR}
 fi
 if [ -z "${SHUNIT_PARENT:-}" ]; then
   _shunit_fatal "zsh does not pass \$0 through properly. please declare \
\"SHUNIT_PARENT=\$0\" before calling shUnit2"
   exit ${SHUNIT_ERROR}
 fi
fi

# shell flags for shunit2:
# u - treat unset variables as an error when performing parameter expansion
__SHUNIT_SHELL_FLAGS='u'

# save the current set of shell flags, and then set some for ourself
shunit_shellFlags_="$-"
for shunit_shellFlag_ in `echo "${__SHUNIT_SHELL_FLAGS}" |sed 's/\(.\)/\1 /g'`
do
 set -${shunit_shellFlag_}
done

#
# constants
#

__SHUNIT_ASSERT_MSG_PREFIX='ASSERT:'
__SHUNIT_PARENT=${SHUNIT_PARENT:-$0}

# set the constants readonly
shunit_constants_=`set |grep "^__SHUNIT_" |cut -d= -f1`
echo "${shunit_constants_}" |grep "^Binary file" >/dev/null
if [ $? -eq 0 ]; then
 # deal with binary junk in 'set' output
 shunit_constants_=`set |grep -a "^__SHUNIT_" |cut -d= -f1`
fi
for shunit_const_ in ${shunit_constants_}; do
 shunit_ro_opts_=''
 if [ -n "${ZSH_VERSION:-}" ]; then
   case ${ZSH_VERSION} in
     [123].*) ;;
     *) shunit_ro_opts_='-g' ;;  # declare readonly constants globally
   esac
 fi
 readonly ${shunit_ro_opts_} ${shunit_const_}
done
unset shunit_const_ shunit_constants_ shunit_ro_opts_
```

```
# variables
__shunit_skip=${SHUNIT_FALSE}
__shunit_suite=''

__shunit_testsPassed=0
__shunit_testsFailed=0
__shunit_testsSkipped=0
__shunit_testsTotal=0

# macros
_SHUNIT_LINENO_='eval if [ "${1:-}" = "--lineno" ]; then [ -n "$2" ] && shunit_message_="[$2]"; shift 2; fi'

#-----------------------------------------------------------------------------
# assert functions
#

#/**
# <s:function group="asserts">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>assertEquals</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>expected</parameter></paramdef>
#       <paramdef>string <parameter>actual</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>Asserts that <emphasis>expected</emphasis> and
#   <emphasis>actual</emphasis> are equal to one another. The message is
#   optional.</para>
# </entry>
# </s:function>
#*/
assertEquals()
{
 ${_SHUNIT_LINENO_}
 if [ $# -lt 2 -o $# -gt 3 ]; then
   _shunit_error 'assertEquals() requires one or two arguments'
   return ${SHUNIT_ERROR}
 fi
 _shunit_shouldSkip && return ${SHUNIT_TRUE}

 [ -z "${shunit_message_:-}" ] && shunit_message_=''
 if [ $# -eq 3 ]; then
```

```
    shunit_message_="${shunit_message_}$1"
    shift
  fi
  shunit_expected_=$1
  shunit_actual_=$2

  shunit_return=${SHUNIT_TRUE}
  if [ "${shunit_expected_}" = "${shunit_actual_}" ]; then
    _shunit_testPassed
  else
    failNotEquals "${shunit_message_}" "${shunit_expected_}" "${shunit_actual_}"
    shunit_return=${SHUNIT_FALSE}
  fi

  unset shunit_message_ shunit_expected_ shunit_actual_ __shunit_lineno
  return ${shunit_return}
}
_ASSERT_EQUALS_='eval assertEquals --lineno "${LINENO:-}"'

#/**
# <s:function group="asserts">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>assertNull</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>value</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>Asserts that <emphasis>value</emphasis> is <literal>null</literal>,
#   or in shell terms a zero-length string. The message is optional.</para>
# </entry>
# </s:function>
#*/
assertNull()
{
  ${_SHUNIT_LINENO_}
  if [ $# -lt 1 -o $# -gt 2 ]; then
    _shunit_error 'assertNull() requires one or two arguments'
    return ${SHUNIT_ERROR}
  fi
  _shunit_shouldSkip && return ${SHUNIT_TRUE}

  [ -z "${shunit_message_:-}" ] && shunit_message_=''
  if [ $# -eq 2 ]; then
```

```
  shunit_message_="${shunit_message_}$1"
  shift
 fi
 if [ $# -eq 2 ]; then
  assertTrue "${shunit_message_}$1" "[ -z '$2' ]"
 else
  assertTrue "[ -z '$1' ]"
 fi
}
_ASSERT_NULL_='eval assertNull --lineno "${LINENO:-}"'

#/**
# <s:function group="asserts">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>assertNotNull</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>value</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>Asserts that <emphasis>value</emphasis> is <emphasis
#   role="strong">not</emphasis> <literal>null</literal>, or in shell terms not
#   a zero-length string. The message is optional.</para>
# </entry>
# </s:function>
#*/
assertNotNull()
{
 ${_SHUNIT_LINENO_}
 if [ $# -gt 2 ]; then  # allowing 0 arguments as $1 might actually be null
  _shunit_error 'assertNotNull() requires one or two arguments'
  return ${SHUNIT_ERROR}
 fi
 _shunit_shouldSkip && return ${SHUNIT_TRUE}

 if [ $# -eq 2 ]; then
  assertTrue "$1" "[ -n '$2' ]"
 else
  assertTrue "[ -n '${1:-}' ]"
 fi
}
_ASSERT_NOT_NULL_='eval assertNotNull --lineno "${LINENO:-}"'

#/**
```

```
# <s:function group="asserts">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>assertSame</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>expected</parameter></paramdef>
#       <paramdef>string <parameter>actual</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function is functionally equivalent to
#   <function>assertEquals</function>.</para>
# </entry>
# </s:function>
#*/
assertSame()
{
 ${_SHUNIT_LINENO_}
 if [ $# -lt 2 -o $# -gt 3 ]; then
   _shunit_error 'assertSame() requires one or two arguments'
   return ${SHUNIT_ERROR}
 fi
 _shunit_shouldSkip && return ${SHUNIT_TRUE}

 if [ $# -eq 2 ]; then
   assertEquals "$1" "$2"
 else
   assertEquals "$1" "$2" "$3"
 fi
}
_ASSERT_SAME_='eval assertSame --lineno "${LINENO:-}"'

#/**
# <s:function group="asserts">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>assertNotSame</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>unexpected</parameter></paramdef>
#       <paramdef>string <parameter>actual</parameter></paramdef>
#     </funcprototype>
```

```
#   </funcsynopsis>
#   <para>Asserts that <emphasis>unexpected</emphasis> and
#   <emphasis>actual</emphasis> are <emphasis role="strong">not</emphasis>
#   equal to one another. The message is optional.</para>
# </entry>
# </s:function>
#*/
assertNotSame()
{
 ${_SHUNIT_LINENO_}
 if [ $# -lt 2 -o $# -gt 3 ]; then
   _shunit_error 'assertNotSame() requires two or three arguments'
   return ${SHUNIT_ERROR}
 fi
 _shunit_shouldSkip && return ${SHUNIT_TRUE}

 [ -z "${shunit_message_:-}" ] && shunit_message_=''
 if [ $# -eq 3 ]; then
   shunit_message_="${shunit_message_}$1"
   shift
 fi
 shunit_unexpected_=$1
 shunit_actual_=$2

 shunit_return=${SHUNIT_TRUE}
 if [ "${shunit_unexpected_}" != "${shunit_actual_}" ]; then
   _shunit_testPassed
 else
   failSame "${shunit_message_}" "$@"
   shunit_return=${SHUNIT_FALSE}
 fi

 unset shunit_message_ shunit_unexpected_ shunit_actual_
 return ${shunit_return}
}
_ASSERT_NOT_SAME_='eval assertNotSame --lineno "${LINENO:-}"'

#/**
# <s:function group="asserts">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>assertTrue</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>condition</parameter></paramdef>
```

```
#     </funcprototype>
#   </funcsynopsis>
#   <para>Asserts that a given shell test condition is true. The message is
#   optional.</para>
#   <para>Testing whether something is true or false is easy enough by using
#   the assertEquals/assertNotSame functions. Shell supports much more
#   complicated tests though, and a means to support them was needed. As such,
#   this function tests that conditions are true or false through evaluation
#   rather than just looking for a true or false.</para>
#   <funcsynopsis>
#     The following test will succeed: <funcsynopsisinfo>assertTrue "[ 34 -gt 23 ]"</funcsynopsisinfo>
#     The folloing test will fail with a message: <funcsynopsisinfo>assertTrue "test failed" "[ -r '/non/existant/file'
]"</funcsynopsisinfo>
#   </funcsynopsis>
# </entry>
# </s:function>
#*/
assertTrue()
{
 ${_SHUNIT_LINENO_}
 if [ $# -gt 2 ]; then
   _shunit_error 'assertTrue() takes one two arguments'
   return ${SHUNIT_ERROR}
 fi
 _shunit_shouldSkip && return ${SHUNIT_TRUE}

 [ -z "${shunit_message_:-}" ] && shunit_message_=''
 if [ $# -eq 2 ]; then
   shunit_message_="${shunit_message_}$1"
   shift
 fi
 shunit_condition_=$1

 # see if condition is an integer, i.e. a return value
 shunit_match_=`expr "${shunit_condition_}" : '\([0-9]*\)'`
 shunit_return=${SHUNIT_TRUE}
 if [ -z "${shunit_condition_}" ]; then
   # null condition
   shunit_return=${SHUNIT_FALSE}
 elif [ "${shunit_condition_}" = "${shunit_match_}" ]; then
   # possible return value. treating 0 as true, and non-zero as false.
   [ ${shunit_condition_} -ne 0 ] && shunit_return=${SHUNIT_FALSE}
 else
   # (hopefully) a condition
   ( eval ${shunit_condition_} ) >/dev/null 2>&1
   [ $? -ne 0 ] && shunit_return=${SHUNIT_FALSE}
 fi
```

```
 # record the test
 if [ ${shunit_return} -eq ${SHUNIT_TRUE} ]; then
   _shunit_testPassed
 else
   _shunit_testFailed "${shunit_message_}"
 fi

 unset shunit_message_ shunit_condition_ shunit_match_
 return ${shunit_return}
}
_ASSERT_TRUE_='eval assertTrue --lineno "${LINENO:-}"'

#/**
# <s:function group="asserts">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>assertFalse</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>condition</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>Asserts that a given shell test condition is false. The message is
#   optional.</para>
#   <para>Testing whether something is true or false is easy enough by using
#   the assertEquals/assertNotSame functions. Shell supports much more
#   complicated tests though, and a means to support them was needed. As such,
#   this function tests that conditions are true or false through evaluation
#   rather than just looking for a true or false.</para>
#   <funcsynopsis>
#     The following test will succeed: <funcsynopsisinfo>assertFalse "[ 'apples' = 'oranges' ]"</funcsynopsisinfo>
#     The folloing test will fail with a message: <funcsynopsisinfo>assertFalse "test failed" "[ 1 -eq 1 -a 2 -eq 2
]"</funcsynopsisinfo>
#   </funcsynopsis>
# </entry>
# </s:function>
#*/
assertFalse()
{
 ${_SHUNIT_LINENO_}
 if [ $# -lt 1 -o $# -gt 2 ]; then
   _shunit_error 'assertFalse() quires one or two arguments'
   return ${SHUNIT_ERROR}
 fi
 _shunit_shouldSkip && return ${SHUNIT_TRUE}
```

```
 [ -z "${shunit_message_:-}" ] && shunit_message_=''
 if [ $# -eq 2 ]; then
   shunit_message_="${shunit_message_}$1"
   shift
 fi
 shunit_condition_=$1

 # see if condition is an integer, i.e. a return value
 shunit_match_=`expr "${shunit_condition_}" : '\([0-9]*\)'`
 shunit_return=${SHUNIT_TRUE}
 if [ -z "${shunit_condition_}" ]; then
   # null condition
   shunit_return=${SHUNIT_FALSE}
 elif [ "${shunit_condition_}" = "${shunit_match_}" ]; then
   # possible return value. treating 0 as true, and non-zero as false.
   [ ${shunit_condition_} -eq 0 ] && shunit_return=${SHUNIT_FALSE}
 else
   # (hopefully) a condition
   ( eval ${shunit_condition_} ) >/dev/null 2>&1
   [ $? -eq 0 ] && shunit_return=${SHUNIT_FALSE}
 fi

 # record the test
 if [ ${shunit_return} -eq ${SHUNIT_TRUE} ]; then
   _shunit_testPassed
 else
   _shunit_testFailed "${shunit_message_}"
 fi

 unset shunit_message_ shunit_condition_ shunit_match_
 return ${shunit_return}
}
_ASSERT_FALSE_='eval assertFalse --lineno "${LINENO:-}"'

#-----------------------------------------------------------------------------
# failure functions
#

#/**
# <s:function group="failures">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>fail</function></funcdef>
```

```
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>Fails the test immediately, with the optional message.</para>
# </entry>
# </s:function>
#*/
fail()
{
 ${_SHUNIT_LINENO_}
 if [ $# -gt 1 ]; then
   _shunit_error 'fail() requires one or two arguments'
   return ${SHUNIT_ERROR}
 fi
 _shunit_shouldSkip && return ${SHUNIT_TRUE}

 [ -z "${shunit_message_:-}" ] && shunit_message_=''
 if [ $# -eq 1 ]; then
   shunit_message_="${shunit_message_}$1"
   shift
 fi

 _shunit_testFailed "${shunit_message_}"

 unset shunit_message_
 return ${SHUNIT_FALSE}
}
_FAIL_='eval fail --lineno "${LINENO:-}"'

#/**
# <s:function group="failures">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>failNotEquals</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>unexpected</parameter></paramdef>
#       <paramdef>string <parameter>actual</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>Fails the test if <emphasis>unexpected</emphasis> and
#   <emphasis>actual</emphasis> are <emphasis role="strong">not</emphasis>
#   equal to one another. The message is optional.</para>
# </entry>
# </s:function>
```

```
#*/
failNotEquals()
{
 ${_SHUNIT_LINENO_}
 if [ $# -lt 2 -o $# -gt 3 ]; then
   _shunit_error 'failNotEquals() requires one or two arguments'
   return ${SHUNIT_ERROR}
 fi
 _shunit_shouldSkip && return ${SHUNIT_TRUE}

 [ -z "${shunit_message_:-}" ] && shunit_message_=''
 if [ $# -eq 3 ]; then
   shunit_message_="${shunit_message_}$1"
   shift
 fi
 shunit_unexpected_=$1
 shunit_actual_=$2

 _shunit_testFailed "${shunit_message_:+${shunit_message_} }expected:<${shunit_unexpected_}> but
was:<${shunit_actual_}>"

 unset shunit_message_ shunit_unexpected_ shunit_actual_
 return ${SHUNIT_FALSE}
}
_FAIL_NOT_EQUALS_='eval failNotEquals --lineno "${LINENO:-}"'

#/**
# <s:function group="failures">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>failSame</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>Indicate test failure because arguments were not the same. The
#   message is optional.</para>
# </entry>
# </s:function>
#*/
failSame()
{
 ${_SHUNIT_LINENO_}
 if [ $# -lt 2 -o $# -gt 3 ]; then
   _shunit_error 'failSame() requires two or three arguments'
```

```
    return ${SHUNIT_ERROR}
  fi
  _shunit_shouldSkip && return ${SHUNIT_TRUE}

  [ -z "${shunit_message_:-}" ] && shunit_message_=''
  if [ $# -eq 3 ]; then
    shunit_message_="${shunit_message_}$1"
    shift
  fi

  _shunit_testFailed "${shunit_message_:+${shunit_message_} }expected not same"

  unset shunit_message_
  return ${SHUNIT_FALSE}
}
_FAIL_SAME_='eval failSame --lineno "${LINENO:-}"'

#/**
# <s:function group="failures">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>failNotSame</function></funcdef>
#       <paramdef>string <parameter>[message]</parameter></paramdef>
#       <paramdef>string <parameter>expected</parameter></paramdef>
#       <paramdef>string <parameter>actual</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>Fails the test if <emphasis>expected</emphasis> and
#   <emphasis>actual</emphasis> are equal to one another. The message is
#   optional.</para>
# </entry>
# </s:function>
#*/
failNotSame()
{
  ${_SHUNIT_LINENO_}
  if [ $# -lt 2 -o $# -gt 3 ]; then
    _shunit_error 'failNotEquals() requires one or two arguments'
    return ${SHUNIT_ERROR}
  fi
  _shunit_shouldSkip && return ${SHUNIT_TRUE}

  if [ $# -eq 2 ]; then
    failNotEquals "$1" "$2"
```

```
 else
   failNotEquals "$1" "$2" "$3"
 fi
}
_FAIL_NOT_SAME_='eval failNotSame --lineno "${LINENO:-}"'

#-----------------------------------------------------------------------------
# skipping functions
#

#/**
# <s:function group="skipping">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>startSkipping</function></funcdef>
#       <paramdef />
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function forces the remaining assert and fail functions to be
#   "skipped", i.e. they will have no effect. Each function skipped will be
#   recorded so that the total of asserts and fails will not be altered.</para>
# </entry>
# </s:function>
#*/
startSkipping()
{
 __shunit_skip=${SHUNIT_TRUE}
}

#/**
# <s:function group="skipping">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>endSkipping</function></funcdef>
#       <paramdef />
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function returns calls to the assert and fail functions to their
#   default behavior, i.e. they will be called.</para>
# </entry>
```

```
# </s:function>
#*/
endSkipping()
{
 __shunit_skip=${SHUNIT_FALSE}
}

#/**
# <s:function group="skipping">
# <entry align="right">
#   <emphasis>boolean</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>isSkipping</function></funcdef>
#       <paramdef />
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function returns the state of skipping.</para>
# </entry>
# </s:function>
#*/
isSkipping()
{
 return ${__shunit_skip}
}

#-----------------------------------------------------------------------------
# suite functions
#

#/**
# <s:function group="suites">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>suite</function></funcdef>
#       <paramdef />
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function can be optionally overridden by the user in their test
#   suite.</para>
#   <para>If this function exists, it will be called when
#   <command>shunit2</command> is sourced. If it does not exist, shUnit2 will
```

```
#   search the parent script for all functions beginning with the word
#   <literal>test</literal>, and they will be added dynamically to the test
#   suite.</para>
# </entry>
# </s:function>
#*/
# Note: see _shunit_mktempFunc() for actual implementation
# suite() { :; }

#/**
# <s:function group="suites">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>suite_addTest</function></funcdef>
#       <paramdef>string <parameter>function</parameter></paramdef>
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function adds a function name to the list of tests scheduled for
#   execution as part of this test suite. This function should only be called
#   from within the <function>suite()</function> function.</para>
# </entry>
# </s:function>
#*/
suite_addTest()
{
 _su_func=${1:-}

 __shunit_suite="${__shunit_suite:+${__shunit_suite} }${_su_func}"

 unset _su_func
}

#/**
# <s:function group="suites">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>oneTimeSetUp</function></funcdef>
#       <paramdef />
#     </funcprototype>
#   </funcsynopsis>
```

```
#   <para>This function can be be optionally overridden by the user in their
#   test suite.</para>
#   <para>If this function exists, it will be called once before any tests are
#   run. It is useful to prepare a common environment for all tests.</para>
# </entry>
# </s:function>
#*/
# Note: see _shunit_mktempFunc() for actual implementation
# oneTimeSetUp() { :; }

#/**
# <s:function group="suites">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>oneTimeTearDown</function></funcdef>
#       <paramdef />
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function can be be optionally overridden by the user in their
#   test suite.</para>
#   <para>If this function exists, it will be called once after all tests are
#   completed. It is useful to clean up the environment after all tests.</para>
# </entry>
# </s:function>
#*/
# Note: see _shunit_mktempFunc() for actual implementation
# oneTimeTearDown() { :; }

#/**
# <s:function group="suites">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>setUp</function></funcdef>
#       <paramdef />
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function can be be optionally overridden by the user in their
#   test suite.</para>
#   <para>If this function exists, it will be called before each test is run.
#   It is useful to reset the environment before each test.</para>
```

```
# </entry>
# </s:function>
#*/
# Note: see _shunit_mktempFunc() for actual implementation
# setUp() { :; }

#/**
# <s:function group="suites">
# <entry align="right">
#   <emphasis>void</emphasis>
# </entry>
# <entry>
#   <funcsynopsis>
#     <funcprototype>
#       <funcdef><function>tearDown</function></funcdef>
#       <paramdef />
#     </funcprototype>
#   </funcsynopsis>
#   <para>This function can be be optionally overridden by the user in their
#   test suite.</para>
#   <para>If this function exists, it will be called after each test completes.
#   It is useful to clean up the environment after each test.</para>
# </entry>
# </s:function>
#*/
# Note: see _shunit_mktempFunc() for actual implementation
# tearDown() { :; }

#------------------------------------------------------------------------------
# internal shUnit2 functions
#

_shunit_cleanup()
{
 name=$1

 case ${name} in
   EXIT) signal=0 ;;
   INT) signal=2 ;;
   TERM) signal=15 ;;
   *)
     _shunit_warn "unrecognized trap value (${name})"
     signal=0
     ;;
 esac

 # do our work
 rm -fr "${__shunit_tmpDir}"
```

```
  # exit for all non-EXIT signals
  if [ ${name} != 'EXIT' ]; then
    _shunit_warn "trapped and now handling the (${name}) signal"
    _shunit_generateReport
    # disable EXIT trap
    trap 0
    # add 128 to signal and exit
    exit `expr ${signal} + 128`
  fi
}

_shunit_execSuite()
{
  echo '#'
  echo '# Performing tests'
  echo '#'
  for _su_func in ${__shunit_suite}; do
    # disable skipping
    endSkipping

    # execute the per-test setup function
    setUp

    # execute the test
    echo "${_su_func}"
    eval ${_su_func}

    # execute the per-test tear-down function
    tearDown
  done

  unset _su_func
}

_shunit_generateReport()
{
  _su__awkPercent='{printf("%4d %3.0f%%", $1, $1*100/$2)}'
  if [ ${__shunit_testsTotal:-0} -gt 0 ]; then
    _su__passed=`echo ${__shunit_testsPassed} ${__shunit_testsTotal} |\
        awk "${_su__awkPercent}"`
    _su__failed=`echo ${__shunit_testsFailed} ${__shunit_testsTotal} |\
        awk "${_su__awkPercent}"`
    _su__skipped=`echo ${__shunit_testsSkipped} ${__shunit_testsTotal} |\
        awk "${_su__awkPercent}"`
    _su__total=`echo ${__shunit_testsTotal} 100 |\
        awk '{printf("%4d %3d%%", $1, $2)}'`
  else
```

```
    _su__passed=`echo 0 0 |awk '{printf("%4d %3d%%", $1, $2)}'`
    _su__failed=${_su__passed}
    _su__skipped=${_su__passed}
    _su__total=${_su__passed}
  fi

  cat <<EOF

#
# Test report
#
tests passed:  ${_su__passed}
tests failed:  ${_su__failed}
tests skipped: ${_su__skipped}
tests total:   ${_su__total}
EOF

  unset _su__awkPercent _su__passed _su__failed _su__skipped _su__total
}

# this function is a cross-platform temporary directory creation tool. not all
# OSes have the mktemp function, so one is included here.
_shunit_mktempDir()
{
  # try the standard mktemp function
  ( exec mktemp -dqt shunit.XXXXXX 2>/dev/null ) && return

  # the standard mktemp didn't work.  doing our own.
  if [ -r '/dev/urandom' ]; then
    _su__random=`od -vAn -N4 -tx4 </dev/urandom |sed 's/^[^0-9a-f]*//'`
  elif [ -n "${RANDOM:-}" ]; then
    # $RANDOM works
    _su__random=${RANDOM}${RANDOM}${RANDOM}$$
  else
    # $RANDOM doesn't work
    _su__date=`date '+%Y%m%d%H%M%S'`
    _su__random=`expr ${_su__date} / $$`
  fi

  _su__tmpDir="${TMPDIR:-/tmp}/shunit.${_su__random}"
  ( umask 077 && mkdir "${_su__tmpDir}" ) || {
    echo 'shUnit:FATAL could not create temporary directory! exiting' >&2
    exit 1
  }

  echo ${_su__tmpDir}
  unset _su__date _su__random _su__tmpDir
}
```

```
# this function is here to work around issues in Cygwin
_shunit_mktempFunc()
{
 for _su__func in oneTimeSetUp oneTimeTearDown setUp tearDown suite; do
   _su__file="${__shunit_tmpDir}/${_su__func}"
   cat <<EOF >"${_su__file}"
#! /bin/sh
exit 0
EOF
   chmod +x "${_su__file}"
 done

 unset _su__file
}

_shunit_shouldSkip()
{
 [ ${__shunit_skip} -eq ${SHUNIT_FALSE} ] && return ${SHUNIT_FALSE}
 _shunit_testSkipped
}

_shunit_testPassed()
{
 __shunit_testsPassed=`expr ${__shunit_testsPassed} + 1`
 __shunit_testsTotal=`expr ${__shunit_testsTotal} + 1`
}

_shunit_testFailed()
{
 _su__msg=$1

 __shunit_testsFailed=`expr ${__shunit_testsFailed} + 1`
 __shunit_testsTotal=`expr ${__shunit_testsTotal} + 1`
 echo "${__SHUNIT_ASSERT_MSG_PREFIX}${_su__msg}" >&2

 unset _su__msg
}

_shunit_testSkipped()
{
 __shunit_testsSkipped=`expr ${__shunit_testsSkipped} + 1`
 __shunit_testsTotal=`expr ${__shunit_testsTotal} + 1`
}

#-----------------------------------------------------------------------------
# main
#
```

```
# create a temporary storage location
__shunit_tmpDir=`_shunit_mktempDir`

# setup traps to clean up after ourselves
trap '_shunit_cleanup EXIT' 0
trap '_shunit_cleanup INT' 2
trap '_shunit_cleanup TERM' 15

# create phantom functions to work around issues with Cygwin
_shunit_mktempFunc
PATH="${__shunit_tmpDir}:${PATH}"

# execute the oneTimeSetUp function (if it exists)
oneTimeSetUp

# execute the suite function defined in the parent test script
# deprecated as of 2.1.0
suite

# if no suite function was defined, dynamically build a list of functions
if [ -z "${__shunit_suite}" ]; then
 shunit_funcs_=`grep "^[ \t]*test[A-Za-z0-9_]* *()" ${__SHUNIT_PARENT} \
     |sed 's/[^A-Za-z0-9_]//g'`
 for shunit_func_ in ${shunit_funcs_}; do
   suite_addTest ${shunit_func_}
 done
fi
unset shunit_func_ shunit_funcs_

# execute the tests
_shunit_execSuite

# execute the oneTimeTearDown function (if it exists)
oneTimeTearDown

# generate report
_shunit_generateReport

# restore the previous set of shell flags
for shunit_shellFlag_ in ${__SHUNIT_SHELL_FLAGS}; do
 echo ${shunit_shellFlags_} |grep ${shunit_shellFlag_} >/dev/null \
   || set +${shunit_shellFlag_}
done
unset shunit_shellFlag_ shunit_shellFlags_

[ ${__shunit_testsFailed} -eq 0 ] || exit 1
```

#/**
</s:shelldoc>
#*/

Found in path(s):
* /opt/cola/permits/1601489760_1679570334.9746232/0/libubox-2022-05-15-d2223ef9-zip/libubox-2022-05-15-d2223ef9/tests/shunit2/shunit2/shunit2

1.38 wolfssl 5.0.0

1.38.1 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.

```
enclave {

   trusted {

   };

   untrusted {
       /* define OCALLs here. */

   };
};
```