OpenSS7 STREAMS Sockets Installation and Reference Manual

Version 0.9.2 Edition 4
Updated 2008-10-31
Package strsock-0.9.2.4

Brian Bidulock <bidulock@openss7.org> for
The OpenSS7 Project <http://www.openss7.org/>

Published by OpenSS7 Corporation
1469 Jefferys Crescent
Edmonton, Alberta T6L 6T1
Canada

This is texinfo edition 4 of the OpenSS7 STREAMS Sockets manual, and is consistent with strsock 0.9.2. This manual was developed under the OpenSS7 Project and was funded in part by OpenSS7 Corporation.

Short Contents

Table of Contents

Preface

Notice

This package is released and distributed under the *AGPL* (see [GNU Affero General Public License], page 102). Please note, however, that there are different licensing terms for the manual pages and some of the documentation (derived from OpenGroup[1] publications and other sources). Consult the permission notices contained in the documentation for more information.

Also note that parts of this software, as identified in Appendix A [Possible BSD Sources], page 99, may be derived from software developed by the *University of California, Berkeley* covered under the *Combined USL BSD License* (see [BSD/USL Combined License], page 113).

This manual is released under the *FDL* (see [GNU Free Documentation License], page 129) with no sections invariant.

Abstract

This manual provides a *Installation and Reference Manual* for *OpenSS7 STREAMS Sockets*.

Objective

The objective of this manual is to provide a guide for the *STREAMS* programmer when developing *STREAMS* modules, drivers and application programs for *OpenSS7 STREAMS Sockets*.

This guide provides information to developers on the use of the *STREAMS* mechanism at user and kernel levels.

STREAMS was incorporated in UNIX System V Release 3 to augment the character input/output (I/O) mechanism and to support development of communication services.

STREAMS provides developers with integral functions, a set of utility routines, and facilities that expedite software design and implementation.

Intent

The intent of this manual is to act as an introductory guide to the *STREAMS* programmer. It is intended to be read alone and is not intended to replace or supplement the *OpenSS7 STREAMS Sockets* manual pages. For a reference for writing code, the manual pages (see `STREAMS(9)`) provide a better reference to the programmer. Although this describes the features of the *OpenSS7 STREAMS Sockets* package, OpenSS7 Corporation is under no obligation to provide any software, system or feature listed herein.

Audience

This manual is intended for a highly technical audience. The reader should already be familiar with *Linux* kernel programming, the *Linux* file system, character devices, driver input and output, interrupts, software interrupt handling, scheduling, process contexts, multiprocessor locks, etc.

[1] Formerly X/Open and UNIX International.

The guide is intended for network and systems programmers, who use the *STREAMS* mechanism at user and kernel levels for *Linux* and *UNIX* system communication services.

Readers of the guide are expected to possess prior knowledge of the *Linux* and *UNIX* system, programming, networking, and data communication.

Revisions

Take care that you are working with a current version of this manual: you will not be notified of updates. To ensure that you are working with a current version, contact the Author, or check The OpenSS7 Project website for a current version.

A current version of this manual is normally distributed with the *OpenSS7 STREAMS Sockets* package, `strsock-0.9.2.4`.[2]

Version Control

```
strsock.texi,v
Revision 0.9.2.16  2008-09-20 11:04:39  brian
- added package patchlevel

Revision 0.9.2.15  2008-08-03 06:03:38  brian
- protected agains texinfo commands in log entries

Revision 0.9.2.14  2008/07/27 08:49:41  brian
- no invariant sections, more libtool ignores

Revision 0.9.2.13  2008-04-28 22:33:30  brian
- updated headers for release

Revision 0.9.2.12  2008-04-25 11:50:54  brian
- updates to AGPLv3

Revision 0.9.2.11  2007/08/12 06:45:02  brian
- updated licenses in manuals

Revision 0.9.2.10  2007/06/23 01:38:27  brian
- updates for release

Revision 0.9.2.9  2007/02/28 06:31:28  brian
- updates and corrections, #ifdef instead of #if

Revision 0.9.2.8  2006/09/18 01:07:16  brian
- updated manuals and release texi docs

Revision 0.9.2.7  2006/09/01 08:55:40  brian
- added headers and working up code

Revision 0.9.2.6  2006/08/29 11:44:05  brian
- added manual pages, working up docs

Revision 0.9.2.5  2006/08/28 10:47:10  brian
- correction

Revision 0.9.2.4  2006/08/28 10:32:58  brian
- updated references
```

[2] http://www.openss7.org/tarballs/strsock-0.9.2.4.tar.bz2

```
Revision 0.9.2.3  2006/08/27 12:27:12  brian
- finalizing auto release files

Revision 0.9.2.2  2006/08/26 09:19:22  brian
- better release file generation

Revision 0.9.2.1  2006/08/23 10:03:59  brian
- started STREAMS Sockets package
```

ISO 9000 Compliance

Only the TEX, texinfo, or roff source for this manual is controlled. An opaque (printed, postscript or portable document format) version of this manual is an **UNCONTROLLED VERSION**.

Disclaimer

OpenSS7 Corporation disclaims all warranties with regard to this documentation including all implied warranties of merchantability, fitness for a particular purpose, non-infringement, or title; that the contents of the manual are suitable for any purpose, or that the implementation of such contents will not infringe on any third party patents, copyrights, trademarks or other rights. In no event shall *OpenSS7 Corporation* be liable for any direct, indirect, special 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 any use of this manual or the performance or implementation of the contents thereof.

OpenSS7 Corporation reserves the right to revise this software and documentation for any reason, including but not limited to, conformity with standards promulgated by various agencies, utilization of advances in the state of the technical arts, or the reflection of changes in the design of any techniques, or procedures embodied, described, or referred to herein. *OpenSS7 Corporation* is under no obligation to provide any feature listed herein.

U.S. Government Restricted Rights

If you are licensing this Software on behalf of the U.S. Government ("Government"), the following provisions apply to you. If the Software is supplied by the Department of Defense ("DoD"), it is classified as "Commercial Computer Software" under paragraph 252.227-7014 of the DoD Supplement to the Federal Acquisition Regulations ("DFARS") (or any successor regulations) and the Government is acquiring only the license rights granted herein (the license rights customarily provided to non-Government users). If the Software is supplied to any unit or agency of the Government other than DoD, it is classified as "Restricted Computer Software" and the Government's rights in the Software are defined in paragraph 52.227-19 of the Federal Acquisition Regulations ("FAR") (or any successor regulations) or, in the cases of NASA, in paragraph 18.52.227-86 of the NASA Supplement to the FAR (or any successor regulations).

Acknowledgements

As with most open source projects, this project would not have been possible without the valiant efforts and productive software of the *Free Software Foundation* and the *Linux Kernel Community*.

Sponsors

Funding for completion of the *OpenSS7 OpenSS7 STREAMS Sockets* package was provided in part by:

- OpenSS7 Corporation

Additional funding for The OpenSS7 Project was provided by:

- OpenSS7 Corporation
- Lockheed Martin Co.
- Motorola
- HOB International
- Comverse Ltd.
- Sonus Networks Inc.
- France Telecom
- SS8 Networks Inc.
- Nortel Networks
- Verisign
- eServGlobal (NZ) Pty Ltd.
- NetCentrex S. A.
- SysMaster Corporation
- GeoLink SA
- AirNet Communications
- TECORE
- Tumsan Oy
- Vodare Ltd.
- Excel Telecommunications

Contributors

The primary contributor to the *OpenSS7 OpenSS7 STREAMS Sockets* package is Brian F. G. Bidulock. The following is a list of significant contributors to The OpenSS7 Project:

- Per Berquist
- John Boyd
- Chuck Winters
- Peter Courtney
- Tom Chandler
- Gurol Ackman
- Kutluk Testicioglu
- John Wenker
- Others

Authors

The authors of the *OpenSS7 OpenSS7 STREAMS Sockets* package include:

– Brian Bidulock

See [Index of Authors], page 145, for a complete listing and cross-index of authors to sections of this manual.

Maintainer

The maintainer of the *OpenSS7 OpenSS7 STREAMS Sockets* package is:

– Brian Bidulock

Please send bug reports to `bugs@openss7.org` using the '`send-pr`' script included in the package, only after reading the '`BUGS`' file in the release, or See Section 7.2 [Problem Reports], page 93.

Web Resources

The OpenSS7 Project provides a website dedicated to the software packages released by the OpenSS7 Project.

Bug Reports

Please send bug reports to `bugs@openss7.org` using the '`send-pr`' script included in the *OpenSS7 STREAMS Sockets* package, only after reading the '`BUGS`' file in the release, or See Section 7.2 [Problem Reports], page 93. You can access the OpenSS7 GNATS database directly via the web, however, the preferred method for sending new bug reports is via mail with the '`send-pr`' script.

Mailing Lists

The OpenSS7 Project provides a number of general discussion Mailing Lists for discussion concerning the *OpenSS7 OpenSS7 STREAMS Sockets* package as well as other packages released by The OpenSS7 Project.

These are `mailman` mailing lists and so have convenient web interfaces for subscribers to control their settings. See `http://www.openss7.org/mailinglist.html`.

The mailing lists are as follows:

'`openss7`' The '`openss7`' mailing list is for general enquiries, information exchange and announcements regarding the OpenSS7 Project. This is our original mailing list and takes the highest amount of traffic.

'`openss7-announce`'
The '`openss7-announce`' mailing list is for announcements related to the OpenSS7 Project. This list will accept announcements posted by subscribers. Subscribe to this list if you are interested in announcements from the OpenSS7 Project, subscribers and sponsors, related to the OpenSS7 Project or STREAMS, SS7, SIGTRAN or SCTP in general.

'`openss7-cvs`'
The '`openss7-cvs`' mailing list is for automatic CVS log reporting. You must get permission of the owner to subscribe to this list. Subscribers are not allowed to post to this list, this is merely for distributing notification of changes to the CVS repository.h

‘openss7-develop’
: The ‘openss7-develop’ mailing list is for email exchange related to the development projects under the OpenSS7 Project. This includes development requests, proposals, requests for comment or proposal. Subscribe to this list if you are interested in ongoing development details regarding the OpenSS7 Project.

‘openss7-test’
: The ‘openss7-test’ mailing list is for email exchange related to the testing of code under the OpenSS7 Project. This specifically relates to conformance testing, verification testing, interoperability testing and beta testing. Subscribe to this list if you are interested in participating in and receiving ongoing details of test activities under the OpenSS7 Project.

‘openss7-bugs’
: The ‘openss7-bugs’ mailing list is specifically tailored to bug tracking. The mailing list takes a feed from the OpenSS7 GNATS bug tracking system and accepts posting of responses to bug reports, tracking and resolution. Subscribe to this list if you are interested in receiving detailed *OpenSS7* release code bug tracking information. This list is not archived; for historical information on problem reports, see our GNATS databases.

‘openss7-updates’
: The ‘openss7-updates’ mailing list provides updates on OpenSS7 Project code releases and ongoing activities. Subscribers are not allowed to post to this list; this list is for official OpenSS7 Project announcements only. Subscribe to this list if you are interested in receiving updates concerning official releases and activities of the OpenSS7 Project.

‘openss7-streams’
: The ‘openss7-streams’ mailing list is for email exchange related to the *STREAMS* development projects under the OpenSS7 Project. This includes development requests, proposals, requests for comment or proposal. Subscribe to this list if you are interested in ongoing development details regarding the OpenSS7 Project *STREAMS* components.

‘linux-streams’
: The ‘linux-streams’ mailing list is for mail exchange related to *Linux Fast-STREAMS* or *Linux STREAMS*. This includes patches, development requests, proposals, requests for comment or proposal. Subscribe to this list if you are interested in ongoing development details regarding the *STREAMS* for Linux components. This is the the new (September 2006) home of the ‘linux-streams’ list formerly of gsyc.escet.urjc.es.

Spam

To avoid spam being sent to the members of the *OpenSS7* mailing list(s), we have blocked mail from non-subscribers. Please subscribe to the mailing list before attempting to post to them. (Attempts to post when not subscribed get bounced.)

As an additional measure against spam, subscriber lists for all *OpenSS7* mailing lists are not accessible to non-subscribers; for most lists subscriber lists are only accessible to the

list administrator. This keeps your mailing address from being picked off our website by bulk mailers.

Acceptable Use Policy

It is acceptable to post professional and courteous messages regarding the *OpenSS7* package or any general information or questions concerning *STREAMS*, *SS7*, *SIGTRAN*, *SCTP* or telecommunications applications in general.

Large Attachments

The mailing list is blocked from messages of greater than `40k`. If you have attachments (patches, test programs, etc.) and you mail them to the list, it will bounce to the list administrator. If you are interested in making your patches, test programs, test results or other large attachments available to the members of the mailing list, state in the message that you would like them posted and the list administrator will place them in the mail archives.

Quick Start Guide

OpenSS7 STREAMS Sockets

Package strsock-0.9.2.4 was released under AGPLv3 2008-10-31.

The *OpenSS7 STREAMS Sockets* package provides *STREAMS* modules and drivers, libraries, programs, initialization scripts, and daemons.

The *OpenSS7 STREAMS Sockets* package contains header files and a number of modules and drivers and the associated documentation originally contained in the *Linux Fast-STREAMS* release, but not contained in the *Linux STREAMS (LiS)* releases.

The package uses the following standard *Open Systems Interconnect (OSI)* conforming header files from the '`strxnet`' package:

- '`<sys/tiuser.h>`' *Transport Provider Interface (TPI)*
- '`<sys/tihdr.h>`' *Transport Provider Interface (TPI)*

The package currently provides the following *STREAMS* modules and drivers:

– '`streams_sockmod.ko`'	*module*
– '`streams_socksys.ko`'	'`/dev/socksys`'

- `sockmod(4)` The `sockmod(4)` module, when pushed on a *TPI Stream*, attempts to provide as wide a range of support for traditional *SVR 4.2* socket module and socket system support.

  It does so by recognizing input-output controls from three sets:
  `sockmod(4)` input-output controls; `socksys(4)` input-output controls, and even `socket(7)` system calls.

- `socksys(4)` The `socksys(4)` driver, when opened, provides a *TPI Stream* that attempts to provide as wide a range of support for traditional *SVR 4.2* socket module and socket system support.

  It does so by recognizing input-output controls from three sets:
  `sockmod(4)` input-output controls; `socksys(4)` input-output controls, and even `socket(7)` system calls.

The package currently provides the following *STREAMS* libraries:

- '`libsocket.so`' '`libsocket.a`'
- '`libsocklib.so`' '`libsocklib.a`'

The '`libsocket`' library implements the following library calls:

`socket(3)`

The '`libsocket`' library only needs to implement the single library call because file descriptors that are returned from the `socket(3)` call have been transformed into real sockets (from the viewpoint of the system call interface) and standard '`glibc`' calls for the remaining functions are used.

The '`libsocklib`' library implements the following library calls:

`accept(3)`	`listen(3)`	`sendto(3)`
`bind(3)`	`recv(3)`	`setsockopt(3)`
`connect(3)`	`recvfrom(3)`	`shutdown(3)`

getpeername(3)	recvmsg(3)	socket(3)
getsockname(3)	send(3)	socketpair(3)
getsockopt(3)	sendmsg(3)	

The '`libsocklib`' library is the older compatibility library approach to providing sockets for *STREAMS* and implements the system calls as library calls for *STREAMS* devices, while calling the '`glibc`' versions for true sockets.

The *OpenSS7 STREAMS Sockets* package includes kernel modules, *SVR 4.2 STREAMS* drivers, modules, libraries, utilities, test programs, daemons, and development environment for the development and execution of *OpenSS7 STREAMS Sockets* modules and drivers.

This distribution is only currently applicable to *Linux* 2.4 and 2.6 kernels and was targeted at `ix86`, `x86_64`, `ppc` and `ppc64` architectures, but should build and install for other architectures as well.

Release

This is the `strsock-0.9.2.4` package, released 2008-10-31. This '`0.9.2.4`' release, and the latest version, can be obtained from the download area of The OpenSS7 Project website using a command such as:

```
$> wget http://www.openss7.org/tarballs/strsock-0.9.2.4.tar.bz2
```

The release is available as an `autoconf(1)` tarball, '`src.rpm`' or '`dsc`', as a set of binary '`rpm`'s or '`deb`'s, or as a `yum(8)` or `apt(8)` repository. See the download page for the `autoconf(1)` tarballs, '`src.rpm`'s, '`dsc`'s, or repository access instructions. See the `strsock` package page for tarballs, source and binary packages.

Please see the '`NEWS`' file for release notes and history of user visible changes for the current version, and the '`ChangeLog`' file for a more detailed history of implementation changes. The '`TODO`' file lists features not yet implemented and other outstanding items.

Please see the '`INSTALL`', '`INSTALL-strsock`' and '`README-make`', files (or see Chapter 6 [Installation], page 43) for installation instructions.

When working from `cvs(1)` or `git(1)`, please see the '`README-cvs`', file (or see Section 6.2.8 [Downloading from CVS], page 55). An abbreviated installation procedure that works for most applications appears below.

See '`README-alpha`' (if it exists) for alpha release information.

Prerequisites

The quickest and easiest way to ensure that all prerequisites are met is to download and install this package from within the *OpenSS7 Master Package*, `openss7-0.9.2.G`, instead of separately.

Prerequisites for the OpenSS7 STREAMS Sockets package are as follows:

1. *Linux* distribution, somewhat *Linux Standards Base* compliant, with a 2.4 or 2.6 kernel and the appropriate tool chain for compiling out-of-tree kernel modules. Most recent

Linux distributions are usable out of the box, but some development packages must be installed. For more information, see Section 5.2 [Compatibility], page 34.

- A fairly LSB compliant GNU/Linux distribution.[1]
- Linux 2.4 kernel (2.4.10 - 2.4.27), or
- Linux 2.6 kernel (2.6.3 - 2.6.26);
- glibc2 or better.
- GNU groff (for man pages).[2]
- GNU texinfo (for info files).

(Note: If you acquired `strsock` a part of the *OpenSS7 Master Package*, then the dependencies listed below will already have been met by unpacking the master package.)

2. *OpenSS7 Linux Fast-STREAMS*, streams-0.9.2.4.[3]
3. *OpenSS7 STREAMS Compatibility Modules*, strcompat-0.9.2.7.
4. *OpenSS7 STREAMS XNS*, strxns-0.9.2.7.
5. *OpenSS7 STREAMS XTI/TLI*, strxnet-0.9.2.12.

When configuring and building multiple *OpenSS7 Project* release packages, place all of the source packages (unpacked tarballs) at the same directory level and all build directories at the same directory level (e.g. all source packages under '`/usr/src`').

When installing packages that install as kernel modules, it is necessary to have the correct kernel development package installed. For the following distributions, use the following commands:

```
Ubuntu:  $> apt-get install linux-headers
Debian:  $> apt-get install kernel-headers
Fedora:  $> yum install kernel-devel
```

You also need the same version of `gcc(1)` compiler with which the kernel was built. If it is not the default, add '`CC=kgcc`' on the line after '`./configure`', for example:

```
$> ../strsock-0.9.2.4/configure CC='gcc-3.4'
```

Installation

The following commands will download, configure, build, check, install, validate, uninstall and remove the package:

```
$> wget http://www.openss7.org/tarballs/strsock-0.9.2.4.tar.bz2
$> tar -xjvf strsock-0.9.2.4.tar.bz2
$> mkdir build
$> pushd build
$> ../strsock-0.9.2.4/configure --enable-autotest
$> make
$> make check
$> sudo make install
$> sudo make installcheck
```

[1] See Section 5.2.1 [GNU/Linux Distributions], page 34, for more information.

[2] If you are using a Debian release, please make sure to install the groff extension package ('`groff_ext`'), as it contains the `refer` or `grefer` commands necessary for including references in the manual pages.

[3] Although, at one time, this package supported *LiS*, *LiS* is now deprecated and unsupported.

```
$> sudo make uninstall
$> popd
$> sudo rm -rf build
$> rm -rf strsock-0.9.2.4
$> rm -f strsock-0.9.2.4.tar.bz2
```

If you have problems, try building with the logging targets instead. If the make of a logging target fails, an automatic problem report will be generated that can be mailed to The OpenSS7 Project.[4] Installation steps using the logging targets proceed as follows:

```
$> wget http://www.openss7.org/tarballs/strsock-0.9.2.4.tar.bz2
$> tar -xjvf strsock-0.9.2.4.tar.bz2
$> mkdir build
$> pushd build
$> ../strsock-0.9.2.4/configure --enable-autotest
$> make compile.log
$> make check.log
$> sudo make install.log
$> sudo make installcheck.log
$> sudo make uninstall.log
$> popd
$> sudo rm -rf build
$> rm -rf strsock-0.9.2.4
$> rm -f strsock-0.9.2.4.tar.bz2
```

See '`README-make`' for additional specialized make targets.

For custom applications, see the '`INSTALL`' and '`INSTALL-strsock`' files or the see Chapter 6 [Installation], page 43, as listed below. If you encounter troubles, see Chapter 7 [Troubleshooting], page 89, before issuing a bug report.

Brief Installation Instructions

The OpenSS7 STREAMS Sockets package is available from the downloads area of The OpenSS7 Project website using a command such as:

```
$> wget http://www.openss7.org/tarballs/strsock-0.9.2.4.tar.bz2
```

Unpack the tarball using a command such as:

```
$> tar -xjvf strsock-0.9.2.4.tar.bz2
```

The tarball will unpack into the relative subdirectory named after the package name: strsock-0.9.2.4.

The package builds using the GNU `autoconf` utilities and the '`configure`' script. To build the package, we recommend using a separate '`build`' directory as follows:

```
$> mkdir build
$> cd build
$> ../strsock-0.9.2.4/configure
```

[4] Please see Section 7.2 [Problem Reports], page 93, or the file '`PROBLEMS`' in the release directory for more information on filing a proper *Problem Report*.

In general, the package configures and builds without adding any special options to the '`configure`' script. For general options to the '`configure`' script, see the GNU '`INSTALL`' file in the distribution:

```
$> less ../strsock-0.9.2.4/INSTALL
```

For specific options to the '`configure`' script, see the '`INSTALL-strsock`' file in the distribution, or simply execute the configure script with the '`--help`' option like so:

```
$> ../strsock-0.9.2.4/configure --help
```

After configuring the package, the package can be compiled simply by issuing the '`make`' command:

```
$> make
```

Some specialized makefile targets exists, see the '`README-make`' file in the distribution or simply invoke the '`help`' target like so:

```
$> make help | less
```

After successfully building the package, the package can be checked by invoking the '`check`' make target like so:

```
$> make check
```

After successfully checking the package, the package can be installed by invoking the '`install`' make target (as root) like so:

```
$> sudo make install
```

The test suites that ship with the package can be invoked after the package has been installed by invoking the '`installcheck`' target. This target can either be invoked as root, or as a normal user, like so:

```
$> make installcheck
```

(Note: you must add the '`--enable-autotest`' flag to '`configure`', above for the test suites to be invoked with '`make installcheck`'.)

The package can be cleanly removed by invoking the '`uninstall`' target (as root):

```
$> sudo make uninstall
```

Then the build directory and tarball can be simply removed:

```
$> cd ..
$> rm -rf build
$> rm -rf strsock-0.9.2.4
$> rm -f strsock-0.9.2.4.tar.bz2
```

Detailed Installation Instructions

More detailed installation instructions can be found in the Chapter 6 [Installation], page 43, contained in the distribution in '`text`', '`info`', '`html`' and '`pdf`' formats:

```
$> cd ../strsock-0.9.2.4
$> less doc/manual/strsock.txt
$> lynx doc/manual/strsock.html
$> info doc/manual/strsock.info
$> xpdf doc/manual/strsock.pdf
```

The '`text`' version of the manual is always available in the '`MANUAL`' file in the release.

The current manual is also always available online from *The OpenSS7 Project* website at:

```
$> lynx http://www.openss7.org/strsock_manual.html
```

1 Introduction

This manual documents the design, implementation, installation, operation and future development schedule of the *OpenSS7 STREAMS Sockets* package.

1.1 Overview

This manual documents the design, implementation, installation, operation and future development of the *OpenSS7 STREAMS Sockets* package.

1.2 Organization of this Manual

This manual is organized (loosely) into several sections as follows:

1.3 Conventions and Definitions

This manual uses *texinfo* typographic conventions.

2 Objective

The objective of the *OpenSS7 STREAMS Sockets* package is to provide *Sockets* capability for *STREAMS* in a similar manner as provided by *SVR 4.2* and later *Solaris* implementations. To accomplish this there are three (3) approaches that can be taken to implementing *Sockets* over *STREAMS*:

2.1 Socket Module

This approach implements the *SVR 4.2* `sockmod(4)` module and cooperating `libsocket(3)` library. The library and cooperating *STREAMS* module behave in a similar fashion to the `libxnet(3)` library and `timod(4)` module for XTI.

There are a set of input-output controls defined for sockets (explicitly for compatibility with *SVR 4.2*) that can be accepted by the `sockmod(4)` module. These input-output controls permit direct conversion between sockets function and kernel-space facilities.

The original `sockmod(4)` approach keep much of the socket state in user space in the fashion of `xti(3)`; however, this approach is just asking for trouble. Also, *Linux* also implements the *SVR 4.2* input-output controls for native Sockets and the resulting `libsocket(3)` library would then work for both *STREAMS*-based sockets and Linux native sockets.

Socket input-output controls defined for *SVR 4.2* are:

O_SI_GETUDATA
SI_SHUTDOWN
SI_LISTEN
SI_SETMYNAME
SI_SETPEERNAME
SI_GETINTRANSIT
SI_TCL_LINK
SI_TCL_UNLINK
SI_SOCKPARAMS
SI_GETUDATA

2.2 Socket System

This approach implements the *SVR 4.2* `socksys(4)` driver. Opening *Streams* via the `socksys(4)` driver causes them to be transformed into native Sockets against which standard '`libc`' socket system calls are sufficient. For compatibility with the `sockmod(4)` approach, the pushing the `sockmod(4)` module can be made to cause the same effect.

Before *Linux Fast-STREAMS* was developed, taking this approach was next to impossible due to the obfuscated nature and non-standard behaviour of the *LiS Stream head*. Now that *LiS* is completely deprecated, this approach is now possible.

SVR 4.2 behaviour for the *Stream head* dictates that it respond to both `termio(7)` and `sockio(7)` input-output controls in addition to
`streamio(7)` controls. That is, each *Stream head* can be a `fifo(4)`, a `pipe(4)`, a `tty(4)`, a `socket` or a `stream`. *LiS* never followed this behaviour: *Linux Fast-STREAMS* was built with it from the start.

Socket input-output controls recognized and intercepted by the *Linux Fast-STREAMS Stream head* are:

SIOCATMARK
SIOCGPGRP
SIOCSPGRP

Additional Sockets input-output controls are passed to the Stream as *TRANSPARENT* or `I_STR(7)` input-output controls.

SIOCHIWAT	Set the high water mark.	*TRANSPARENT*
SIOGHIWAT	Get the high water mark.	*TRANSPARENT*
SIOCLOWAT	Set the low water mark.	*TRANSPARENT*
SIOGLOWAT	Get the low water mark.	*TRANSPARENT*
SIOCATMARK	Test whether at mark.	*Stream head*
SIOCGPGRP	Get process group.	*Stream head*
FIOCGPGRP		
SIOCSPGRP	Set process group.	*Stream head*
FIOCSPGRP		
FIONREAD TIOCINQ	Number of bytes to be read.	`I_STR(7)`
TIOOUTQ	Number of bytes to be sent.	`I_STR(7)`
FIONBIO	Set blocking i/o.	*TRANSPARENT*
SIOCPROTO	Set a new protocol entry.	*TRANSPARENT*
SIOCGETNAME	Get socket's name.	*TRANSPARENT*
SIOCGETPEER	Get socket's peer.	*TRANSPARENT*
SIOCXPROTO	Clear protocol entry table.	*TRANSPARENT*
SIOCSOCKSYS	(See table below.)	*TRANSPARENT*

Command	**Library Call**	**System Call**
SO_SOCKET	`socket(3)`	–
SO_ACCEPT	`accept(3)`	`accept(2)`
SO_BIND	`bind(3)`	`bind(2)`
SO_CONNECT	`connect(3)`	`connect(2)`
SO_GETPEERNAME	`getpeername(3)`	`getpeername(2)`
SO_GETSOCKNAME	`getsockname(3)`	`getsockname(2)`
SO_GETSOCKOPT	`getsockopt(3)`	`getsockopt(2)`
SO_LISTEN	`listen(3)`	`listen(2)`
SO_RECV	`recv(3)`	`recv(2)`
SO_RECVFROM	`recvfrom(3)`	`recvfrom(2)`
SO_SEND	`send(3)`	`send(2)`
SO_SENDTO	`sendto(3)`	`sendto(2)`
SO_SETSOCKOPT	`setsockopt(3)`	`setsockopt(2)`
SO_SHUTDOWN	`shutdown(3)`	`shutdown(2)`
SO_GETIPDOMAIN	–	`getipdomain(2)`
SO_SETIPDOMAIN	–	`setipdomain(2)`
SO_SETREUID	–	`setreuid(2)`
SO_SETREGID	–	`getreuid(2)`
SO_GETTIME	–	`gettime(2)`
SO_SETTIME	–	`settime(2)`
SO_GETITIMER	–	`getitimer(2)`
SO_SETITIMER	–	`setitimer(2)`

SO_SELECT	–	`select(2)`
SO_ADJTIME	–	`adjtimex(2)`
SO_RECVMSG	`recvmsg(3)`	`recvmsg(2)`
SO_SENDMSG	`sendmsg(3)`	`sendmsg(2)`
SO_SOCKPAIR	`socketpair(3)`	`socketpair(2)`

2.3 Socket Integration

With this approach, *Streams* that are usable for sockets are given the *IF_SOCK* attribute during registration. The *Stream head* is aware of this and creates a socket structure for each *Stream head* opened for the driver.

2.4 Solaris Approaches

The following is reprinted from an article in *UNIX® Insider* issue '`3/9/01`' written by *Jim Mauro*.

> Prior to *Solaris 2.6*, sockets were an abstraction that existed at the library level. That is, much of the socket state and socket semantics support were provided with the '`libsocket`' library. The kernel's view of a process's socket connection entailed a file descriptor and linkage to a *Stream head*, which provided the path to the underlying transport. The disparity between the library socket state and the kernel's view was one of several reasons a new implementation was introduced in *Solaris 2.6*.
>
> To provide a relevant basis for comparison, we will start by looking at what happens in the pre-*Solaris 2.6* release (this is, releases up to and including *Solaris 2.5.1*) when a socket is created. The major software layers are shown in Figure XXX for reference.
>
> The primary software components are the socket library and the `sockmod(4)` *STREAMS* module. The `specfs(5)` layer is shown for completeness and is part of the layering, due to the use of pseudo-devices as an entry point into the networking layers. To digress for a moment, the special file system, `specfs(5)` came out of *SVR 4 UNIX®* as a means of addressing the issue of *device special files* that exist on *UNIX®* on-disk file systems (e.g. *UFS*). *UNIX®* system have always abstracted I/O (input/output) devices through device special files. The '`/dev`' directory name space stores files that represent physical devices and pseudo-devices on the system. Using device major numbers, those device files provide an entry point into the appropriate device driver, and using minor numbers, they are able to uniquely identify one of potentially many devices of the same type. (That is something of an oversimplification, but it is sufficient for our purpose here in describing `specfs(5)`).
>
> The '`/dev`' directory resides on the '`root`' file system, which is an instance of *UFS*. As such, references to the file system and its files and directories are handed using the *UFS* file system operations and *UFS* file operations. That is usually sufficient, but not desired behaviour for device special files. I/O to a device special file requires entry into a device driver. That is, issuing an `open(2)` system call on '`/dev/rmt/0`' means someone wishes to open the tape device represented by '`/dev/rmt/0`', thereby entering the appropriate driver's

`xx_open()` routine. As a file on a *UFS* file system, the typical open routine called would be the `ufs_open()` code, but that is not what we want for devices. The '`specfs(5)`' file system was designed to address such situations; it provides a straightforward mechanism for linking the underlying structures for file support in the kernel to the required device driver interfaces. Like all file systems in *Solaris* (and any *SVR4*-based *UNIX*®) it is based on the *VFS*/`vnode` infrastructure. (See *Solaris Internals* and *UNIX*® *Internals* in the Resources section for detailed information on *VFS*.)

Getting back to sockets in *Solaris 2.5.1*, the `specfs(5)` layer comes into play because the socket open ultimately results in an `open(2)` system call issued on the `tcp(4)` or `udp(4)` pseudo-device. More precisely, the socket library passes the arguments given to the `socket(3)` call to a lookup function that searches an internal (internal to '`libsocket.so`') array to match the domain argument and retrieve a corresponding character string. It then uses the character string to find a match in the '`/etc/netconfig`' file, which is used for transport selection and describes all the available transport protocols in *Solaris*. (See `netconfig(5)`.) This transport selection mechanism is an essential part of a network programming implementation; it allows for the interfaces to be protocol-independent, so the programmer is not required to maintain a different source base for Ethernet-based networks versus FDDI-based networks, etc.

A `netconfig(5)` data structure (defined in '`/usr/include/sys/netconfig.h`') is populated based on the line entries in '`/etc/netconfig`' that match the domain (as per the character string retrieved from the internal table), type, and protocol family specified in the `socket(3)` call. Among the `netconfig(5)` parameters, a device is defined that provides the entry point into the transport provider kernel module. For example, a call to '`socket(AF_INET, SOCK_STREAM, 0)`' indicates an Internet transport that provides reliable, connection-oriented behaviour is desired. The *TCP* layer of the *TCP/IP* protocol family provides such a service, and the '`/etc/netconfig`' entry defines '`/dev/tcp`' as the device to open for entry into that transport layer. The socket library code will issue an `open(2)` on '`/dev/tcp`' accordingly. If one were developing a network-based application using the *X/Open Transport Interface (XTI)* – a superset of what was the *Transport Layer Interface (TLI)* – the `t_open(3)` call would receive the '`/dev/tcp`' argument explicitly for a connection using *TCP* as a transport protocol.

The block sitting below the `specfs(5)` in Figure XXX, the *Stream head*, is a generic part of a *STREAMS*-based communication path. The *Stream head* is created when a *STREAMS* device is opened. In Figure XXX, the `open(2)` to the '`/dev/tcp`' transport layer, which is a *STREAMS* device, resulted in the creation of the *Stream head*. The *Stream head* translates the interface calls made by the socket library into *STREAMS* messages (the *STREAMS* framework is message-based and uses queues to move data downstream [from the user process to the *STREAMS* driver] and upstream [from the driver to the user process]). The *STREAMS* facility provides for the insertion (pushing) and removal (popping) of *STREAMS* modules in the data flow, between the *Stream*

head and the underlying driver. Each module implements a set of queues – a read queue and a write queue – for processing the data and messages. The generic picture is shown in Figure YYY.

In the context of *Solaris 2.5.1* sockets, the *STREAMS* module shown in Figure YYY is a kernel `sockmod(4)` module (located in the '`/kernel/strmod`' directory). `sockmod(4)` provides, in conjunction with '`libsocket.so`', support for socket semantics using the *STREAMS* facility. That is, socket calls are handled initially by the socket library, then passed down to the *Stream head*, which transforms the calls into *STREAMS* messages and passes them down to `sockmod(4)`. Upstream messages are passed from the underlying device driver and transport provider through `sockmod(4)` and back up to the process. Thus, the functions contained in the `sockmod(4)` module include *STREAMS* queue reading and writing in the form of queue '`read put`' and '`write put`' code for moving data up and down the *Stream* as data is read and written from the socket. The `sockmod(4)` module communicates with the underlying transport using primitives and structures defines in the '`/usr/include/sys/tihdr.h`' header file.

The socket state maintained at the library level is in the form of a library-internal data structure, `_si_user`, which maintains various bits of information about the socket, and is what the internal socket create function returns on a socket call. Yes, it is the file descriptor that represents the socket that is returned to the user code `_si_user` is visible only to the library. You will find the structure definition for `_si_user` and associated structures that it links to (`si_udata` and `si_sockparams`) in '`/usr/include/sys/sockmod.h`'. If you look at the structure definition, you will see that the `_si_user` embeds the `si_udata` and `si_sockparams` structures, which maintain stat information (e.g. connected, bound), socket options (accept connection), information on the transport provider (e.g. service type), and family, type, and protocol used for the socket.

At the `sockmod(4)` layer, a socket is internally represented in the `so_so` data structure. Fields of interest there include an embedded `ti_info` structure ('`/usr/include/sys/tiuser.h`') that manages transport provider information, a network buffer (`netbuf`) for data transfer, a `si_udata` structure that replicates the socket state (among other things), and message blocks (`mblk_t`), which are the basic unit of communication across *STREAMS*.

In *Solaris 2.6*, we did away with the `sockmod(4)` *STREAMS* module and trimmed a lot of code from '`libsocket`'. Most of the socket-related library interfaces result in system call traps into the kernel, without any library-level code executing. A few of the interfaces (`socket(3)` and `sockpair(3)`) execute some library-level code before entering the kernel. However, all the state information is maintained in the kernel, where it belongs. This creates a nice visibility feature – we can now see file descriptors that represent sockets.

```
sunsys> uname -a
SunOS sunsys 5.8 Generic_108528-01 sun4u sparc SUNW, Ultra-60
sunsys> srv &
[1]     7153
Socket port: # 34940
```

```
Send bug: 16384, Rcv buf: 24576

sunsys> pfiles 7153
7153:   srv
  Current rlimit: 1024 file descriptors
   0: S_IFCHR mode:0620 dev:32,0 ino:91176 uid:19822 gid:7 rdev:24,14
      O_RDWR|O_LARGEFILE
   1: S_IFCHR mode:0620 dev:32,0 ino:91176 uid:19822 gid:7 rdev:24,14
      O_RDWR|O_LARGEFILE
   2: S_IFCHR mode:0620 dev:32,0 ino:91176 uid:19822 gid:7 rdev:24,14
      O_RDWR|O_LARGEFILE
   3: S_IFSOCK mode:0666 dev:186,0 ino:63137 uid:0 gid:0 size:0
      O_RDWR
        sockname: AF_INET 0.0.0.0  port: 34940
sunsys>
```

In the above example, a simple *TCP* socket server process is started (`srv`, PID 7153). (The '`Socket port`' and '`Send buf`' lines are output from the `srv` process when it starts.) Using the `pfiles(1)` command to dump the process's open file descriptors, we see that the file descriptor is identified as a socket, and we even get the socket type (*AF_INET*) and port number. (The freeware command, `lsof(1)`, is a great utility for extracting process file descriptor information if you are on an older *Sun OS* that does not have `pfiles(1)`. You can get `lsof(1)` from `ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/`.)

The '`libsocket`' changes associated with `sockfs(5)` maintain the documented interfaces. Both source and binary compatibility are maintained, as socket code compiled on early version of *Solaris* should work without recompilation on *Solaris 6* and later releases. Source code should move over and recompile with no changes as well.

The trimming down of the library-level socket code required providing a new means to map the domain type passed as an argument to `socket(3)` to facility lookup in '`/etc/netconfig`'. Recall that the *Solaris 2.5.1* socket library did this using an internal table. In *Solaris 6* and later, a new configuration file and command is introduced to provide that functionality. The '`/etc/sock2path`' contains the necessary information to map the `socket(3)` call parameters to the appropriate transport provider and device. A new command, `soconfig(8)`, is used to maintain '`/etc/sock2path`'. It is executed automatically at boot time via an entry in the '`/etc/inittab`' file. Reference the `sock2path(4)` and `soconfig(8)` manual pages for specifics. For most applications, the default entries in `sock2path(5)` are sufficient.

As a file system (pseudo-file system), `sockfs(5)` implements the generic *VFS*/`vnode` related support structures and exports the required file-system-specific functions. However, the entry into the `sockfs(5)`-specific functions does not necessarily follow the typical flow of a regular file open, which is vectored to the file-type-specific function through the use of macros and an operations table. That is, the issuing of an `open(2)` system call on a file enters a generic `vnode` code path and ultimately resolves through a `VOP_OPEN()` macro to the appropriate file-system-specific open code (e.g. `ufs_open` for a file an a *UFS* file system).

Sockets are created an opened using the socket(3) API. A call to socket(3) from user code enters the 'libsocket' library, which handles the mapping to the transport provider device, then enters the sockfs(5) kernel module through an internal so_create() system call. The sock_open() (file system specific open routine) is invoked through the so_create() call, which is how other necessary create functions, such as an initialization function for the socket *Stream*, are called.

Other conventional system calls, such as read(2) or write(2) on a socket, are vectored into the sockfs(5) specific read and write code (sock_read() and sock_write()) through the standard *VFS*/vnode mechanism. Once entered, the sockfs(5) read/write code makes lower-level calls into the sockfs(5) subsystem designed to interface with the transport provider. For example, a read(2) system call on a socket vectors into sock_read(), which does some basic housekeeping and calls an internal sorecvmsg() (socket receive message) function. In sorecvmsg(), socket stat is tested and the request is moved downstream via a call to the *STREAMS* get-message function.

The most compelling part of the sockfs(5) implementation is that consolidation of all socket stat information is in a single structure, maintained in one place: the kernel. Sockets are represented internally as a sonode, defined in '/usr/include/sys/socketvar.h'. All operations on a sonode take place within the kernel sockfs(5) subsystem, isolating state changes and eliminating the need to replicate state for consistency.

3 Reference

3.1 Files

STRSOCK creates the following kernel modules files in the kernel modules directory, '`/lib/modules/2.4.20-28.7/`':[1]

'`modules.strsock`'

STRSOCK installs the following kernel module files in the kernel modules directory, '`/lib/modules/2.4.20-28.7/strsock/`':[2]

'`streams_sockmod.ko`'
'`streams_socksys.ko`'

STRSOCK installs the following header files in the system include directory, '`/usr/include/strsock/`':

'`sys/strsock/config.h`'
'`sys/strsock/version.h`'
'`sys/sockio.h`'
'`sys/sockmod.h`'
'`sys/sockpath.h`'
'`sys/socksys.h`'
'`sys/socklib.h`'
'`sockdb.h`'
'`sockio.h`'
'`sockmod.h`'
'`socksys.h`'

STRSOCK installs the following test programs in the system libexec directory, '`/usr/libexec/strsock/`':[3]

'`send-pr`'
'`send-pr.config`'
: The `send-pr` stand-alone shell script can be used for the automatic generation of problem reports for the *OpenSS7 STREAMS Sockets* package. The '`send-pr.config`' file provides localized definitions used by the `send-pr` program. For more information on problem reports, See Section 7.2 [Problem Reports], page 93, and, in particular, See Section 7.2.4 [Stand Alone Problem Reports], page 96.

'`test-sockpath`'
: test case executable.

'`test-socket`'
: test case executable.

'`test-socklib`'
: test case executable.

[1] The kernel version '`2.4.20-28.7`' is just and example. For the running kernel, '`uname -r`' is expected.
[2] The kernel version '`2.4.20-28.7`' is just and example. For the running kernel, '`uname -r`' is expected.
[3] Note that on some systems, '`/usr/libexec`' does not exist, and '`/usr/lib`' is used instead.

‘test-sockmod’
: test case executable.

‘test-socksys’
: test case executable.

‘testsuite’
‘atlocal’
: The testsuite stand-alone shell script invokes test cases in the test programs above as compiled into a comprehensive regression, troubleshooting and validation test suite for the *OpenSS7 STREAMS Sockets* drivers. The ‘atlocal’ file provides localized definitions used by the testsuite program. For more information on test suites, See Section 7.1 [Test Suites], page 89, and, in particular, See Section 7.1.2.1 [Running Test Suites], page 93.

STRSOCK installs the following utility programs in the system binary directory, ‘/usr/sbin/’:

‘initsock’
‘soconfig’
‘strsock_mknod’
: This utility can be used by init scripts or administrative users to create or remove device nodes in the ‘/dev’ directory for *OpenSS7 STREAMS Sockets* drivers.

STRSOCK installs the following static and shared object libraries in the system library directory, ‘/usr/lib/’ (32-bit libraries) and ‘/usr/lib64/’ (64-bit libraries):

‘libsockpath.so.0.0.0’
‘libsockpath.so.0’
‘libsockpath.so’
‘libsockpath.a’
‘libsockpath.la’
‘libsocket.a’
‘libsocket.la’
‘libsocklib.a’
‘libsocklib.la’

STRSOCK installs the following init scripts in the system init directory, ‘/etc/rc.d/init.d/’ (non-Debian) or ‘/etc/init.d/’ (Debian):

‘strsock’
: This is the name of the system init script on non-Debian based systems.

‘strsock.sh’
: This is the name of the system init script on Debian based systems.

STRSOCK installs the following system configuration files in the configuration directory, ‘/etc/’:

‘strsock.conf’
: This file provided configuration information for any system controls affected by the ‘strsock’ package.

‘modutils/strsock’
: This file provides module definitions and demand loading aliases for the ‘strsock’ package. This file is really only applicable to older 2.4 kernels.

‘`sock2path`’
: This file provides the `sock2path(5)` definitions for the *OpenSS7 STREAMS Sockets* drivers. `sock2path(5)` definitions are used by the ‘`strsock`’ package and the ‘`libsocket`’ library.

STRSOCK installs the following system configuration file in the system configuration directory, ‘`/etc/sysconfig/`’ (non-Debian) or ‘`/etc/default/`’ (Debian):

‘`strsock`’
: This file provides system configuration information used by init scripts for the ‘`strsock`’ package. Some options of init script execution can be controlled by this file.

STRSOCK installs the following info files in the system info directory, ‘`/usr/share/info/`’:

‘`strsock.info`’
‘`strsock.info-1`’
‘`strsock.info-2`’
: These files contain this manual in *GNU info* format.

STRSOCK installs the following manual page macros and reference database files in the system man directory, ‘`/usr/share/man/`’:[4]

‘`strsock.macros`’
: This file contains manual page macro definitions included by the manual pages included in the package.

‘`strsock.refs`’
: This file contains a reference database referenced by the manual pages included in the package.

STRSOCK installs the following manual pages in the system man directory, ‘`/usr/share/man/man3/`’:

‘`accept.3`’
‘`bind.3`’

‘`connect.3`’
‘`getpeername.3`’
‘`getsockname.3`’
‘`getsockopt.3`’
‘`libsocket.3`’
‘`listen.3`’
‘`recv.3`’

‘`recvfrom.3`’
‘`recvmsg.3`’
‘`send.3`’

[4] Note that macro and reference databse files are not installed if the package is configured for cooked manpages.

‘sendmsg.3’
‘sendto.3’
‘setsockopt.3’
‘shutdown.3’
‘socket.3’
‘socketpair.3’
‘sockets.3’
‘endsockpath.3’
‘freesockpathent.3’
‘getsockpath.3’
‘getsockpathent.3’
‘setsockpath.3’
‘sockpath.3’
‘sp_perror.3’
‘sp_sperror.3’
‘libsocklib.3’
‘socklib.3’

STRSOCK installs the following manual pages in the system man directory, ‘/usr/share/man/man4/’:

‘sockmod.4’
‘socksys.4’

STRSOCK installs the following manual pages in the system man directory, ‘/usr/share/man/man5/’:

‘strsock.5’
: manual page for the strsock(5) package.

‘sock2path.5’

STRSOCK installs the following manual pages in the system man directory, ‘/usr/share/man/man8/’:

‘strsock_mknod.8’
: Documentation for the strsock_mknod(8) utility program.

‘initsock.8’
‘soconfig.8’
‘test-socket.8’
‘test-socklib.8’
‘test-sockmod.8’
‘test-sockpath.8’
‘test-socksys.8’

3.2 Drivers

3.3 Modules

3.4 Libraries

3.5 Utilities

3.6 Development

4 Conformance

5 Releases

This is the OpenSS7 Release of the OpenSS7 STREAMS Sockets tools, drivers and modules used with the *Linux Fast-STREAMS* or *Linux STREAMS*[1] SVR 4.2 STREAMS releases.

The purpose of providing a separate release of this package was to separate the OpenSS7 STREAMS Sockets tools, headers, drivers and modules from the *Linux STREAMS*[2] package for use with both *Linux STREAMS*[3] and *Linux Fast-STREAMS* in preparation for replacement of the former by the later.

The following sections provide information on OpenSS7 STREAMS Sockets releases as well as compatibility information of OpenSS7 release to mainstream UNIX releases of the core, modules and drivers, as well as Linux kernel compatibility.

5.1 Prerequisites

The quickest and easiest way to ensure that all prerequisites are met is to download and install this package from within the *OpenSS7 Master Package*, `openss7-0.9.2.G`, instead of separately.

Prerequisites for the OpenSS7 STREAMS Sockets package are as follows:

1. *Linux* distribution, somewhat *Linux Standards Base* compliant, with a 2.4 or 2.6 kernel and the appropriate tool chain for compiling out-of-tree kernel modules. Most recent *Linux* distributions are usable out of the box, but some development packages must be installed. For more information, see Section 5.2 [Compatibility], page 34.
   - A fairly LSB compliant GNU/Linux distribution.[4]
   - Linux 2.4 kernel (2.4.10 - 2.4.27), or
   - Linux 2.6 kernel (2.6.3 - 2.6.26);
   - glibc2 or better.
   - GNU groff (for man pages).[5]
   - GNU texinfo (for info files).

(Note: If you acquired `strsock` a part of the *OpenSS7 Master Package*, then the dependencies listed below will already have been met by unpacking the master package.)

2. *OpenSS7 Linux Fast-STREAMS*, streams-0.9.2.4.[6]
3. *OpenSS7 STREAMS Compatibility Modules*, strcompat-0.9.2.7.
4. *OpenSS7 STREAMS XNS*, strxns-0.9.2.7.
5. *OpenSS7 STREAMS XTI/TLI*, strxnet-0.9.2.12.

If you need to rebuild the package from sources with modifications, you will need a larger GNU tool chain as described in See Section 6.2.8 [Downloading from CVS], page 55.

---

[1] *Linux STREAMS* is buggy, unsupported and deprecated. Do not use it.

[2] *Linux STREAMS* is buggy, unsupported and deprecated. Do not use it.

[3] *Linux STREAMS* is buggy, unsupported and deprecated. Do not use it.

[4] See Section 5.2.1 [GNU/Linux Distributions], page 34, for more information.

[5] If you are using a Debian release, please make sure to install the groff extension package (‘`groff_ext`’), as it contains the `refer` or `grefer` commands necessary for including references in the manual pages.

[6] Although, at one time, this package supported *LiS*, *LiS* is now deprecated and unsupported.

5.2 Compatibility

This section discusses compatibility with major prerequisites.

5.2.1 GNU/Linux Distributions

OpenSS7 STREAMS Sockets is compatible with the following *Linux* distributions:[7]

- CentOS Enterprise Linux 3.4 (centos34) TBD
- CentOS Enterprise Linux 4.0 (centos4) TBD
- CentOS Enterprise Linux 4.92 (centos49) TBD
- CentOS Enterprise Linux 5.0 (centos5)
- CentOS Enterprise Linux 5.1 (centos51)
- CentOS Enterprise Linux 5.2 (centos52)
- Debian 3.0r2 Woody (deb3.0) TBD
- Debian 3.1r0a Sarge (deb3.1) TBD
- Debian 4.0r1 Etch (deb4.0)
- Debian 4.0r2 Etch (deb4.0)
- Debian 4.0r3 Etch (deb4.0)
- Fedora Core 1 (FC1) TBD
- Fedora Core 2 (FC2) TBD
- Fedora Core 3 (FC3) TBD
- Fedora Core 4 (FC4) TBD
- Fedora Core 5 (FC5) TBD
- Fedora Core 6 (FC6) TBD
- Fedora 7 (FC7)
- Fedora 8 (FC8)
- Fedora 9 (FC9)
- Gentoo 2006.1 (untested) TBD
- Gentoo 2007.1 (untested) TBD
- Lineox 4.026 (LEL4) TBD
- Lineox 4.053 (LEL4) TBD
- Mandrakelinux 9.2 (MDK92) TBD
- Mandrakelinux 10.0 (MDK100) TBD
- Mandrakelinux 10.1 (MDK101) TBD
- Mandriva Linux LE2005 (MDK102) TBD
- Mandriva Linux LE2006 (MDK103) TBD
- Mandriva One (untested)
- RedHat Linux 7.2 (RH7)
- RedHat Linux 7.3 (RH7)

[7] Items marked as '`TBD`' are scheduled to have support deprecated. That is, in a future release, the distributions marked '`TBD`' will not longer be validated before release.

- RedHat Linux 8.0 (RH8) TBD
- RedHat Linux 9 (RH9) TBD
- RedHat Enterprise Linux 3.0 (EL3) TBD
- RedHat Enterprise Linux 4 (EL4)
- RedHat Enterprise Linux 5 (EL5)
- SuSE 8.0 Professional (SuSE8.0) TBD
- SuSE 9.1 Personal (SuSE9.1) TBD
- SuSE 9.2 Professional (SuSE9.2) TBD
- SuSE OpenSuSE (SuSEOSS) TBD
- SuSE 10.0 (SuSE10.0) TBD
- SuSE 10.1 (SuSE10.1) TBD
- SuSE 10.2 (SuSE10.2) TBD
- SuSE 10.3 (SuSE10.3) TBD
- SuSE 11.0 (SuSE11.0)
- SLES 9 (SLES9) TBD
- SLES 9 SP2 (SLES9) TBD
- SLES 9 SP3 (SLES9) TBD
- SLES 10 (SLES10)
- Ubuntu 5.10 (ubu5.10) TBD
- Ubuntu 6.03 LTS (ubu6.03) TBD
- Ubuntu 6.10 (ubu6.10) TBD
- Ubuntu 7.04 (ubu7.04) TBD
- Ubuntu 7.10 (ubu7.10)
- Ubuntu 8.04 (ubu8.04)
- WhiteBox Enterprise Linux 3.0 (WBEL3) TBD
- WhiteBox Enterprise Linux 4 (WBEL4) TBD

When installing from the tarball (see Section 6.5.3 [Installing the Tar Ball], page 76), this distribution is probably compatible with a much broader array of distributions than those listed above. These are the distributions against which the current maintainer creates and tests builds.

5.2.2 Kernel

The *OpenSS7 STREAMS Sockets* package compiles as a *Linux* kernel module. It is not necessary to patch the *Linux* kernel to build or use the package.[8] Nor do you have to recompile your kernel to build or use the package. OpenSS7 packages use `autoconf` scripts to adapt the package source to your existing kernel. The package builds and runs nicely against production kernels from the distributions listed above. Rather than relying on kernel versions, the `autoconf` scripts interrogate the kernel for specific features and variants to

[8] At a later date, it is possible to move this package into the kernel, however, with continued resistance to STREAMS from within the *Linux* developer community, this is currently unlikely.

better adapt to distribution production kernels that have had patches applied over the official kernel.org sources.

The *OpenSS7 STREAMS Sockets* package is compatible with 2.4 kernel series after 2.4.10 and has been tested up to and including 2.4.27. It has been tested from 2.6.3 up to and including 2.6.26 (with Fedora 9, openSUSE 11.0 and Ubuntu 8.04 patchsets). Please note that your mileage may vary if you use a kernel more recent than 2.6.26.4: it is difficult to anticipate changes that kernel developers will make in the future. Many kernels in the 2.6 series now vary widely by release version and if you encounter problems, try a kernel within the supported series.

UP validation testing for kernels is performed on all supported architectures. SMP validation testing was initially performed on UP machines, as well as on an Intel 3.0GHz Pentium IV 630 with HyperThreading enabled (2x). Because HyperThreading is not as independent as multiple CPUs, SMP validation testing was limited. Current releases have been tested on dual 1.8GHz Xeon HP servers (2x) as well as dual quad-core SunFire (8x) servers.

It should be noted that, while the packages will configure, build and install against XEN kernels, that problems running validation test suites against XEN kernels has been reported. *XEN kernels are explicitly not supported.* This may change at some point in the future if someone really requires running OpenSS7 under a XEN kernel.

5.2.3 Architectures

The *OpenSS7 STREAMS Sockets* package compiles and installs on a wide range of architectures. Although it is believed that the package will work on all architectures supported by the Linux kernel being used, validation testing has only been performed with the following architectures:

- ix86
- x86_64
- ppc (MPC 860)
- ppc64

32-bit compatibility validation testing is performed on all 64-bit architectures supporting 32-bit compatibility. If you would like to validate an OpenSS7 package on a specific machine architecture, you are welcome to sponsor the project with a test machine.

5.2.4 Linux STREAMS

The *OpenSS7 STREAMS Sockets* package is currently compatible with *Linux STREAMS*,[9] however, to use the *OpenSS7 STREAMS Sockets* package with *LiS* requires use of the OpenSS7 release packages of *LiS*. The *OpenSS7 STREAMS Sockets* package is compatible with the OpenSS7 **LiS-2.18.7** release that is available from the The OpenSS7 Project Downloads Page. But, do not use *LiS*: it is buggy, unsupported and deprecated. Use *Linux Fast-STREAMS* instead.

5.2.5 Linux Fast-STREAMS

The *OpenSS7 STREAMS Sockets* package is currently compatible with *Linux Fast-STREAMS (LfS)*. The *OpenSS7 STREAMS Sockets* package is compatible with

[9] *Linux STREAMS* is buggy, unsupported and deprecated. Do not use it.

the OpenSS7 **streams-0.9.2.4** release that is available from the The OpenSS7 Project Downloads Page.

5.3 Release Notes

The sections that follow provide information on OpenSS7 releases of the *OpenSS7 STREAMS Sockets* package.

Major changes for release strsock-0.9.2.4

This is the fourth separate *OpenSS7 Project* release of the *OpenSS7 STREAMS Sockets* package whose components were formerly part of the '`strss7`', '`strxnet`' and '`streams`' packages. These networking modules provide the Sockets library. The package contains all the necessary manual pages and other documentation. The package is packaged as an OpenSS7 style autoconf tarball.

This is the fourth alpha release and is part of the OpenSS7 Master Package ('`openss7-0.9.2.G`'). The release includes maintenance support for recent distributions and tool chain, but also includes some performance and feature upgrades and inspection bug fixes. It deprecates previous releases. Please upgrade before reporting bugs on previous releases.

Major features since the last public release are as follows:

- Minor documentation corrections.
- Kernel module license made explicit "GPL v2". And then changed back to "GPL".
- License upgrade to *AGPL Version 3*.
- Support for flex 2.5.33 in maintainer mode.
- Ability to strap out major documentation build and installation primarily for embedded targets.
- Improvements to common build process for embedded and cross-compile targets.
- Updated tool chain to '`m4-1.4.12`', '`autoconf-2.63`' and '`texinfo-4.13`'.
- Conversion of RPM spec files to common approach for major subpackages.
- Updated references database for manual pages and roff documents.
- Build system now builds `yum(8)` repositories for RPMs and `apt-get(8)` repositories for DEBs. Installation documentation has been updated to include details of repository install sourcesref.
- Added `MODULE_VERSION` to all modules and drivers.
- Several important bug fixes for thread safety of the socket library: see '`BUGS`' in the release for more information.

This is a public alpha release of the package: it deprecates previous releases. Please upgrade to the current release before reporting bugs. Please see '`README-alpha`' in the release, or *Section 5.4 [Maturity], page 39.*

As with other OpenSS7 releases, this release configures, compiles, installs and builds RPMs and DEBs for a wide range of Linux 2.4 and 2.6 RPM- and DPKG-based distributions, and can be used on production kernels without patching or recompiling the kernel.

This package is publicly released under the *GNU Affero General Public License Version 3*. The release is available as an `autoconf` tarball, SRPM, DSC, and set of binary RPMs and

DEBs. See the downloads page for the `autoconf` tarballs, SRPMs and DSCs. For tarballs, SRPMs, DSCs and binary RPMs and DEBs, see the strsock package page.

See `http://www.openss7.org/codefiles/strsock-0.9.2.4/ChangeLog` and `http://www.openss7.org/codefiles/strsock-0.9.2.4/NEWS` in the release for more information. Also, see the '`strsock.pdf`' manual in the release (also in html `http://www.openss7.org/strsock_manual.html`).

For the news release, see `http://www.openss7.org/rel20081029_C.html`.

Major changes for release strsock-0.9.2.3

This *OpenSS7 Project* release is the third separate release of the *OpenSS7 STREAMS Sockets* package whose components were formerly part of the '`strss7`', '`strxnet`' and '`streams`' packages. These networking modules provide the Sockets library. The package contains all the necessary manual pages and other documentation. The package is packaged as an OpenSS7 style autoconf tarball.

This is the third alpha release and is part of the OpenSS7 Master Package ('`openss7-0.9.2.F`'). This release is primarily a maintenance release supporting recent distributions and tool chains.

Major features since the last public release are as follows:

- Support build on openSUSE 10.2.
- Support build on Fedora 7 with 2.6.21 kernel.
- Support build on CentOS 5.0 (RHEL5).
- Support build on Ubuntu 7.04.
- Updated to gettext 0.16.1.
- Changes to support build on 2.6.20-1.2307.fc5 and 2.6.20-1.2933.fc6 kernel.
- Supports build on Fedora Core 6.
- Support for recent distributions and tool chains.

Major changes for release strsock-0.9.2.2

This *OpenSS7 Project* release is the second separate release of the *OpenSS7 STREAMS Sockets* package whose components were formerly part of the '`strss7`', '`strxnet`' and '`streams`' packages. These networking modules provide the Sockets library. The package contains all the necessary manual pages and other documentation. The package is packaged as an OpenSS7 style autoconf tarball.

This is the second alpha release for *Linux Fast-STREAMS* and is part of the OpenSS7 Master Package ('`openss7-0.9.2.E`'). This release is primarily a maintenance release supporting recent distributions and tool chains.

Major features since the last public release are as follows:

- Improvements to the common build environment with better support for standalone package builds on 2.4 kernels.
- Support for autoconf 2.61, automake 1.10 and gettext 0.16.
- Support for Ubuntu 6.10 distribution and bug fixes for i386 kenels.
- The package now looks for other subpackages with a version number as unpacked by separate tarball.

Major changes for release strsock-0.9.2.1

This *OpenSS7 Project* release is the first separate release of the *OpenSS7 STREAMS Sockets* package whose components were formerly part of the '`strss7`', '`strxnet`' and '`streams`' packages. These networking modules provide the Sockets library. The package contains all the necessary manual pages and other documentation. The package is packaged as an OpenSS7 style autoconf tarball.

This is the initial alpha release for *Linux Fast-STREAMS* and is part of the OpenSS7 Master Package ('`openss7-0.9.2.D`').

This release is the initial public release.

- Support for most recent 2.6.18 kernels (including Fedora Core 5 with inode diet patch-set).

5.4 Maturity

The *OpenSS7 Project* adheres to the following release philosophy:

- pre-alpha release
- alpha release
- beta release
- gamma release
- production release
- unstable release

5.4.1 Pre-Alpha Releases

Pre-alpha releases are releases that have received no testing whatsoever. Code in the release is not even known to configure or compile. The purpose of a pre-alpha release is to make code and documentation available for inspection only, and to solicit comments on the design approach or other characteristics of the software package.

Pre-alpha release packages ship containing warnings recommending that the user not even execute the contained code.

5.4.2 Alpha Releases

Alpha releases are releases that have received little to no testing, or that have been tested and contains known bugs or defects that make the package unsuitable even for testing. The purpose for an *alpha* release are the same as for the pre-alpha release, with the additional purpose that it is an early release of partially functional code that has problems that an external developer might be willing to fix themselves and contribute back to the project.

Alpha release packages ship containing warnings that executing the code can crash machines and might possibly do damage to systems upon which it is executed.

5.4.3 Beta Releases

Beta releases are releases that have received some testing, but the testing to date is not exhaustive. *Beta* release packages do not ship with known defects. All known defects are resolved before distribution; however, as exhaustive testing has not been performed,

unknown defects may exist. The purpose for a *beta* release is to provide a baseline for other organizations to participate in the rigorous testing of the package.

Beta release packages ship containing warnings that the package has not been exhaustively tested and that the package may cause systems to crash. Suitability of software in this category for production use is not advised by the project; however, as always, is at the discretion of the user of the software.

5.4.4 Gamma Releases

Gamma releases are releases that have received exhaustive testing within the project, but external testing has been minimal. *Gamma* release packages do not ship with known defects. As exhaustive internal testing has been performed, unknown defects should be few. Please remember that there is NO WARRANTY on public release packages.

Gamma release packages typically resolve problems in previous *beta* releases, and might not have had full regression testing performed. Suitability of software in this category for production use is at the discretion of the user of the software. *The OpenSS7 Project* recommends that the complete validation test suites provided with the package be performed and pass on target systems before considering production use.

5.4.5 Production Releases

Production releases are releases that have received exhaustive testing within the project and validated on specific distributions and architectures. *Production* release packages do not ship with known defects. Please remember that there is NO WARRANTY on public release packages.

Production packages ship containing a list of validated distributions and architectures. Full regression testing of any maintenance changes is performed. Suitability of software in this category for production use on the specified target distributions and architectures is at the discretion of the user. It should not be necessary to preform validation tests on the set of supported target systems before considering production use.

5.4.6 Unstable Releases

Unstable releases are releases that have received extensive testing within the project and validated on a a wide range of distributions and architectures; however, is has tested unstable and found to be suffering from critical problems and issues that cannot be resolved. Maintenance of the package has proved impossible. *Unstable* release packages ship with known defects (and loud warnings). Suitability of software in this category for production use is at the discretion of the user of the software. *The OpenSS7 Project* recommends that the problems and issues be closely examined before this software is used even in a non-production environment. Each failing test scenario should be completely avoided by the application. *OpenSS7* beta software is more stable that software in this category.

5.5 Bugs

5.5.1 Defect Notices

OpenSS7 STREAMS Sockets has unknown defects. This is an *alpha* release. Some defects could be harmful. No validation testing whatsoever has been performed by the *OpenSS7*

Project on this software. The software might fail to configure or compile on some systems. The *OpenSS7 Project* recommends that you **do not use this software for purposes other than develoment or evaluation, and then only with great care**. Use at your own risk. Remember that there is **NO WARRANTY**.[10]

This software is *alpha* software. As such, it can likely crash your kernel. Installation of the software can irreparably mangle your header files or Linux distribution in such a way as to make it unusable. Crashes could lock your system and rebooting the system might not repair the problem. You can possibly lose all the data on your system. Because this software can crash your kernel, the resulting unstable system could destroy computer hardware or peripherals making them unusable. You could void the warranty on any system on which you run this software. YOU HAVE BEEN WARNED.

5.5.2 Known Defects

With the exception of packages not originally created by the *OpenSS7 Project*, the *OpenSS7 Project* software does not ship with known bugs in any release stage except *pre-alpha*. *OpenSS7 STREAMS Sockets* had no known bugs at the time of release.

5.5.3 Defect History

This section contains historical bugs that were encountered during development and their resolutions. This list serves two purposes:

1. It captures bugs encountered between releases during development that could possibly reoccur (and the Moon is made of blue cheese). It therefore provides a place for users to look if they encounter a problem.
2. It provides a low overhead bug list between releases for developers to use as a '`TODO`' list.

Bugs

`001. 2008-08-17T03:42:23+0000`

The socket, socklib and sockpath libraries were discovered to contain a thread-safety bug caused by newer behaviour of pthread_once() causing the libraries to core dump when used on recent implementations of pthreads (nptl).

fixed in `strsock-0.9.2.4`.

5.6 Schedule

Current Plan

This package is still incomplete. It is missing a fully functional Stream head socket library, module or driver implementation. Also, performance testing of STREAMS INET drivers indicates that STREAMS-based networking components could exhibit far superior performance to that exhibited by legacy Linux sockets. To accomplish this requires fully implementing `socklib(3)`, `sockmod(4)` module or `socksys(4)` driver, creating validation test suites, and performance analysis and comparison with legacy Linux mechanisms.

[10] See sections **Disclaimer of Warranty** and **Limitation of Liability** under [GNU Affero General Public License], page 102.

This package is not currently a priority for the OpenSS7 project, which is focussed on more telecom-specific protocol stacks and capabilities. Nevertheless, if completion of this package is important to your industry and your organization is able to fund further development or contribute the missing items, contact us on the openss7-develop maling list. Until funding or a mandate surfaces, this package will likely continue as a proof-of-concept only. It will be maintained ina compilable and installable package (that is, it will be updated for current Linux distributions and kernels) on the same basis as other packages in the OpenSS7 master package.

Things to Do

- Testing. This package is completely untested.
  todo
- Create a `socksys(4)` *Socket System STREAMS* driver.
  todo
- It would be interesting to perform some *Netperf* and *Iperf* performance tests against *STREAMS*-based network drivers supporting *Transport Provider Interface (TPI)* and *Sockets* using this socket system. Without `sockmod(4)` and '`libsocket`' it was only possible to use *Netperf* for performance testing, and then only using the XTI tests.
  todo
- Create the `socket(3)` '`libsocket`' shared library.
  todo
- Create a `sockmod(4)` *Socket Module STREAMS* module.
  todo
- Move already written code from the '`stacks`' or '`strss7`' directories into the '`src`' directory.
  todo
- Create a skeleton directory and manual and place `strsock` as a sub-package in the *OpenSS7 Master Package*.
  done
  You are reading it.

The `strsock` package is currently incomplete.

The purpose of the package was to move *STREAMS* socket capabilities outside of the *STREAMS* release package so as to allow them to be used with both *LiS* and *Linux Fast-STREAMS*. As *Linux Fast-STREAMS* has already shown superior to *LiS* in terms of both performance and conformance, it is no longer necessary to keep these packages separate and they can be combined once again (or still) with *Linux Fast-STREAMS*.

5.7 History

For the latest developments with regard to history of changes, please see the '`ChangeLog`' file in the release package.

6 Installation

6.1 Repositories

The OpenSS7 STREAMS Sockets package release can be accessed from the repositories of The OpenSS7 Project. For `rpm(1)` based systems, the package is available in a `yum(8)` repository based on '`repomd`' XML and may also be accessed using `zypper(8)` or `yast(8)`. For `dpkg(1)` based systems, the package is available in a `apt(8)` repository.

By far the easiest (most repeatable and manageable) form for installing and using *OpenSS7* packages is to install packages from the `yum(8)` or `apt(8)` repositories. If your distribution does not support `yum(8)`, `zypper(8)`, `yast(8)` or `apt(8)`, then it is still possible to install the RPMs or DEBs from the repositories using `rpm(1)`, `dpkg(1)`; or by using `wget(1)` and then installing them from RPM or DEB using `rpm(1)` or `dpkg(1)` locally.

If binaries are not available for your distribution or specific kernel, but your distribution supports `rpm(1)` or `dpkg(1)`, the next best method for installing and using *OpenSS7* packages is to download and rebuild the source RPMs or DSCs from the repository. This can also be performed with `yum(8)`, `zypper(8)`, `yast(8)`, `apt(8)`; or directly using `wget(1)`, `rpm(1)` or `dpkg(1)`.

If your architecture does not support `rpm(1)` or `dpkg(1)` at all, or you have special needs (such as cross-compiling for embedded targets), the final resort method is to download, configure, build and install from tarball. In this later case, the easiest way to build and install *OpenSS7* packages from tarball is to use the tarball for the *OpenSS7 Master Package*, `openss7-0.9.2.G`.

6.1.1 Repositories for YUM

To install or upgrade from the *OpenSS7* '`repomd`' repositories, you will need a file in your '`/etc/yum.repo.d/`' directory. This file can be obtained directly from the *OpenSS7* repository, like so:

```
$> REPOS="http://www.openss7.org/repos/rpms"
$> wget $REPOS/centos/5.2/x86_64/repodata/openss7.repo
$> sudo cp -f openss7.repo /etc/yum.repo.d/
$> sudo yum makecache
```

This example assumes the the distribution is '`centos`' and the distribution release is '`5.2`' and the architecture requires is '`x86_64`'. Another example would be '`$REPOS/i686/suse/11.0/i686/repodata/openss7.repo`', for using `yum(8)` with SUSE.

Once the repository is set up, *OpenSS7* includes a number of virtual package definitions that eas the installation and removal of kernel modules, libraries and utilities. Downloading, configuring, building and installation for a single-kernel distribution is as easy as:

```
$> sudo yum install strsock
```

Removing the package is as easy as:

```
$> sudo yum remove strsock
```

If you have difficulty downloading the '`openss7.repo`' file, edit the following information into the file and place it into the '`/etc/yum.repo.d/openss7.repo`' file:

```
-| [openss7]
-| enabled = 1
-| name = OpenSS7 Repository
-| baseurl = http://www.openss7.org/repos/rpms/centos/5.2/x86_64
-| gpgcheck = 1
-| gpgkey = http://www.openss7.org/pubkey.asc
```

Note that it is also possible to point to these repositories as an additional installation source when installing CentOS, RedHat, Fedora, or others. You will have an additional *STREAMS* category from which to choose installation packages.

Some additional installation real or virtual package names and the installations they accomplish are as follows:

‘`strsock`’
: This package can be used to install or remove the entire OpenSS7 STREAMS Sockets package. When installing, kernel modules will be installed automatically for the highest version kernel on your system. When removing, all corresponding kernel modules will also be removed.

‘`strsock-devel`’
: This package can be used to install or remove the development components of the OpenSS7 STREAMS Sockets package. When installing, ‘`strsock`’ and appropriate kernel module and kernel module development and debug packages will also be installed. When removing, the development package and all kernel module development and debug packages will also be removed.

‘`strsock-2.4.20-28.7`’
: This package can be used to install or remove the package for a specific kernel version. When installing, the ‘`strsock`’ package will also be installed if necessary. When removing the last kernel module package, the ‘`strsock`’ package will also be removed.

  Note that the version ‘`2.4.20-28.7`’ is just an example. Use the version returned by ‘`$(uname -r)`’ for the kernel for which you wish to install or remove the packages.

‘`strsock-2.4.20-28.7-devel`’
: This package can be used to install or remove the development and debug packages for a specific kernel version. When installing, the ‘`strsock`’ and ‘`strsock-devel`’ packages will also be installed if necessary. When removing the development and debug for kernel modules for the last kernel, the ‘`strsock-devel`’ package will also be removed.

  Note that the version ‘`2.4.20-28.7`’ is just an example. Use the version returned by ‘`$(uname -r)`’ for the kernel for which you wish to install or remove the packages.

For assistance with specific RPMs, see Section 6.2.3 [Downloading the Binary RPM], page 47.

6.1.2 Repositories for APT

For assistance with specific DEBs, see Section 6.2.4 [Downloading the Debian DEB], page 50.

6.2 Downloading

The OpenSS7 STREAMS Sockets package releases can be downloaded from the downloads page of The OpenSS7 Project. The package is available as a binary RPM (for popular architectures) a source RPM, Debian binary DEB and source DSC, or as a tar ball. If you are using a browsable viewer, you can obtain the OpenSS7 release of `strsock` from the links in the sections that follow.

By far the easiest (most repeatable and manageable) form for installing and using *OpenSS7* packages is to download and install individual packages from binary RPM or DEB. If binary RPMs or DEBs are not available for your distribution, but your distribution supports `rpm(1)` or `dpkg(1)`, the next best method for installing and using *OpenSS7* packages is to download and rebuild the source RPMs or DSCs.

If your architecture does not support `rpm(1)` or `dpkg(1)` at all, or you have special needs (such as cross-compiling for embedded targets), the final resort method is to download, configure, build and install from tarball. In this later case, the easiest way to build and install *OpenSS7* packages from tarball is to use the tarball for the *OpenSS7 Master Package*, `openss7-0.9.2.G`.

6.2.1 Downloading with YUM

OpenSS7 repositories support `yum(8)` and `zypper(8)` in repomd XML format as well as YaST and YaST2 formats.

OpenSS7 includes virtual packages that ease the installation and removal of kernel modules, libraries and utilities. Downloading, configuration, building and installation for a signle-kernel distribution installation is as easy as:

```
% sudo yum install strsock
```

This and additional packages for installation are detailed as follows:

‘`strsock`’ Install this package if you need the runtime ‘`strsock`’ package.

```
% sudo yum install strsock
```

This will install the ‘`strsock`’, ‘`strsock-lib`’ and ‘`strsock-KVERSION`’ RPMs, where ‘`KVERSION`’ is the highest version number kernel on your system.

Remove this package if you need to remove all vestages of the ‘`strsock`’ package.

```
% sudo yum remove strsock
```

This will remove the ‘`strsock`’, ‘`strsock-lib`’, ‘`strsock-devel`’, ‘`strsock-KVERSION`’ and ‘`strsock-devel-KVERSION`’ RPMs for all kernels on your system.

‘`strsock-devel`’

Install this package if you need the development ‘`strsock`’ package.

```
% sudo yum install strsock-devel
```

This will install the ‘`strsock`’, ‘`strsock-lib`’, ‘`strsock-devel`’, ‘`strsock-KVERSION`’ and ‘`strsock-devel-KVERSION`’ RPMs, where ‘`KVERSION`’ is the highest version number kernel on your system.

Remove this package if you do not need development capabilities for the ‘`strsock`’ package for any kernel.

```
% sudo yum remove strsock-devel
```

This will remove the '`strsock-devel`' and '`strsock-devel-KVERSION`' RPMs for all kernels on your system.

'`strsock-2.4.20-28.7`'

Install this package if you need the runtime '`strsock`' for kernel version '`2.4.20-28.7`'. The value '`2.4.20-28.7`' is just an example. For the running kernel, you can install the runtime '`strsock`' components with:

```
% sudo yum install strsock-$(uname -r)
```

This will install the '`strsock`', '`strsock-lib`' and '`strsock-2.4.20-28.7`' RPMs, where '`2.4.20-28.7`' is the kernel version specified.

Remove this package if you no longer need the runtime '`strsock`' for kernel version '`2.4.20-28.7`'. The value '`2.4.20-28.7`' is just an example. For the running kernel, you can remove the runtime '`strsock`' components with:

```
% sudo yum remove strsock-$(uname -r)
```

This will remove the '`strsock-2.4.20-28.7`' and '`strsock-devel-2.4.20-28.7`' RPMs, where '`2.4.20-28.7`' is the kernel version specified. Also, if this is the last kernel for which '`strsock`' was installed, the '`strsock`' '`strsock-lib`' and '`strsock-devel`' RPMs will also be removed.

Note that this is a virtual package name: the actual RPMs installed or removed from the system is a kernel module package whose precise name will depend upon the system being used.

'`strsock-devel-2.4.20-28.7`'

Install this package if you need the development '`strsock`' package for kernel version '`2.4.20-28.7`'. The value '`2.4.20-28.7`' is just an example. For the running kernel, you can install the kernel development '`strsock`' components with:

```
% sudo yum install strsock-devel-$(uname -r)
```

This will install the '`strsock`', '`strsock-lib`', '`strsock-devel`', '`strsock-2.4.20-28.7`' and '`strsock-devel-2.4.20-28.7`' RPMs, where '`2.4.20-28.7`' is the kernel version specified.

Remove this package if you no longer need the development capabilities for the '`strsock`' package for kernel version '`2.4.20-28.7`'. The value '`2.4.20-28.7`' is just an example. For the running kernel, you can remove the kernel development '`strsock`' components with:

```
% sudo yum remove strsock-devel-$(uname -r)
```

This will remove the '`strsock-devel-2.4.20-28.7`' RPMs, where '`2.4.20-28.7`' is the kernel version specified. Also, if this is the last kernel for which '`strsock`' was installed, the '`strsock-devel`' RPMs will also be removed.

Note that this is a virtual package name: the actual RPMs installed or removed from the system is a kernel module package whose precise name will depend upon the system being used.

‘strsock-lib’
: This package is an auxillary package that should be removed and inserted automatically by yum(8). In rare instances you might need to remove or install this package explicitly.

6.2.2 Downloading with APT

OpenSS7 repositries support apt(8) repositorie digests and signatures.

6.2.3 Downloading the Binary RPM

To install from binary RPM, you will need several of the RPM for a complete installation. Binary RPM fall into several categories. To download and install a complete package requires the appropriate RPM from each of the several categories below, as applicable. Some release packages do not provide RPMs in each of the several categories.

To install from Binary RPM, you will need all of the following kernel independent packages for your architecture, and one of the kernel-dependent packages from the next section.

Independent RPM

Independent RPM are dependent on neither the Linux kernel version, nor the *STREAMS* package. For example, the source package ‘strsock-source-0.9.2.4-1.7.2.noarch.rpm’, is not dependent on kernel nor *STREAMS* package.

All of the following kernel and *STREAMS* independent RPM are required for your architecture. Binary RPMs listed here are for example only: additional binary RPMs are available from the downloads site. If your architecture is not available, you can build binary RPM from the source RPM (see see Section 6.4.1 [Building from the Source RPM], page 73).

Architecture Independent

strsock-dev-0.9.2.4-1.7.2.noarch.rpm
: The ‘strsock-dev’ package contains the device definitions necessary to run applications programs developed for OpenSS7 STREAMS Sockets.[1]

strsock-doc-0.9.2.4-1.7.2.noarch.rpm
: The ‘strsock-doc’ package contains this manual in plain text, postscript, ‘pdf’ and ‘html’ forms, along with the meta-information from the ‘strsock’ package. It also contains all of the manual pages necessary for developing OpenSS7 STREAMS Sockets applications and OpenSS7 STREAMS Sockets *STREAMS* modules or drivers.

strsock-init-0.9.2.4-1.7.2.noarch.rpm
: The ‘strsock-init’ package contains the init scripts and provides the ‘postinst’ scripts necessary to create kernel module preloads and modules definitions for all kernel module ‘core’ subpackages.

[1] Not all distributions support the ‘%dev’ RPM macro: a case in point is the SuSE 8.0 distribution which uses an older version of rpm(1). Distributions that do not support the ‘%dev’ macro will build devices as a ‘%post’ operation. Note also that not all release packages contain devices. Only packages that provide *STREAMS* character device drivers need devices, and then only when the ‘specfs’ or ‘devfsd’ is not being used.

strsock-source-0.9.2.4-1.7.2.noarch.rpm
: The '`strsock-source`' package contains the source code necessary for building the OpenSS7 STREAMS Sockets release. It includes the `autoconf(1)` configuration utilities necessary to create and distribute tarballs, '`rpm`' and '`deb`'/'`dsc`'.[2]

Architecture Dependent

strsock-devel-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-devel`' package contains library archives for static compilation, header files to develop OpenSS7 STREAMS Sockets modules and drivers. This also includes the header files and static libraries required to compile OpenSS7 STREAMS Sockets applications programs.

strsock-lib-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-lib`' package contains the run-time shared libraries necessary to run application programs and utilities developed for the '`strsock`' package.[3]

STREAMS-Dependent RPM

STREAMS-Dependent RPM are dependent upon the specific *STREAMS* package being used, either *Linux STREAMS* or *Linux Fast-STREAMS*. Packages dependent upon *Linux STREAMS* will have '`LiS`' in the package name. Packages dependent upon *Linux Fast-STREAMS* will have '`streams`' in the package name. Note that some *STREAMS*-Dependent RPM are also Kernel-Dependent RPM as described below.

One of the following *STREAMS*-Dependent packages is required for your architecture. If your architecture is not on the list, you can build binary RPM from the source RPM (see see Section 6.4.1 [Building from the Source RPM], page 73).

strsock-LiS-util-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-LiS-util`' package provides administrative and configuration test utilities and commands associated with the OpenSS7 STREAMS Sockets package. Because this package must link a *STREAMS*-specific library, it is a *STREAMS*-Dependent package. Use the '`strsock-LiS-util`' package if you have *LiS* installed.

strsock-streams-util-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-streams-util`' package provides administrative and configuration test utilities and commands associated with the OpenSS7 STREAMS Sockets package. Because this package must link a *STREAMS*-specific library, it is a *STREAMS*-Dependent package. Use the '`strsock-streams-util`' package if you have streams installed.

Kernel-Dependent RPM

Kernel-Dependent RPM are dependent on specific Linux Kernel Binary RPM releases. Packages are provided for popular released *RedHat* kernels. Packages dependent upon *Red-*

[2] Note that not all releases have source RPM packages. Release packages that do not contain kernel modules do not generate a source RPM package.

[3] Note that not all release packages contain shared libraries, and, therefore, not all release packages contain this package.

Hat or other kernel RPM will have the '`_kversion`' kernel package version in the package name.

One of the following Kernel-Dependent packages is required for your architecture and kernel version. If your architecture or kernel version is not on the list, you can build binary RPM from the source RPM (see see Section 6.4.1 [Building from the Source RPM], page 73).[4]

strsock-core-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-core`' package contains the loadable kernel modules that depend only on the kernel. This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version '`2.4.20-28.7`'.[5]

strsock-info-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-info`' package[6] contains the module symbol version information for the '`core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loading the actual kernel modules (from the '`core`' subpackage above). This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version '`2.4.20-28.7`'.[7]

strsock-LiS-core-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-LiS-core`' package contains the kernel modules that provide the OpenSS7 STREAMS Sockets *STREAMS* modules and drivers. This package is heavily tied to the *STREAMS* package and kernel for which it was compiled. This particular package applies to '`LiS`' (*Linux STREAMS*) on kernel version '`2.4.20-28.7`'.[8]

strsock-streams-core-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-streams-core`' package contains the kernel modules that provide the OpenSS7 STREAMS Sockets *STREAMS* modules and drivers. This package is heavily tied to the *STREAMS* package and kernel for which it was compiled. This particular package applies to '`streams`' (*Linux Fast-STREAMS*) on kernel version '`2.4.20-28.7`'.[9]

strsock-LiS-info-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-LiS-info`' package[10] contains the module symbol version information for the '`LiS-core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loaded the

---

[4] Note that on *Mandrakelinux*, unlike other RPM kernel distributions, kernel packages for the ix86 architectures are always placed in i586 architecture packages regardless of the true processor architecture of the kernel package. '`configure`' detects this and builds the appropriate packages.

[5] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example. Note also that only release packages that contain kernel modules will contain a '`core`' subpackage.

[6] Note that only release packages that contain kernel modules and that export versioned symbols will contain a '`info`' subpackage. Also, this subpackage is only applicable to 2.4 series kernels and is not necessary and not built for 2.6 series kernels.

[7] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[8] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[9] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[10] Note that only release packages that contain kernel modules and that export versioned symbols will contain a '`LiS-info`' subpackage.

actual kernel modules (from the '`LiS-core`' subpackage above). This package is heavily tied to the *STREAMS* package and kernel for which it was compiled. This particular package applies to '`LiS`' (*Linux STREAMS*) on kernel version '`2.4.20-28.7`'.[11]

strsock-streams-info-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm
: The '`strsock-streams-info`' package[12] contains the module symbol version information for the '`streams-core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loaded the actual kernel modules (from the '`streams-core`' subpackage above). This package is heavily tied to the *STREAMS* package and kernel for which it was compiled. This particular package applies to '`streams`' (*Linux Fast-STREAMS*) on kernel version '`2.4.20-28.7`'.[13]

Configuration and Installation

To configure, build and install the binary RPM, See Section 6.3.1 [Configuring the Binary RPM], page 58.

6.2.4 Downloading the Debian DEB

To install from binary DEB, you will need several of the DEB for a complete installation. Binary DEB fall into several categories. To download and install a complete package requires the appropriate DEB from each of the several categories below, as applicable. Some release packages do not provide DEBs in each of the several categories.

To install from Binary DEB, you will need all of the following kernel independent packages for your architecture, and one of the kernel-dependent packages from the next section.

Independent DEB

Independent DEB are dependent on neither the Linux kernel version, nor the *STREAMS* package. For example, the source package '`strsock-source_0.9.2.4-0_i386.deb`', is not dependent on kernel nor *STREAMS* package.

All of the following kernel and *STREAMS* independent DEB are required for your architecture. Binary DEBs listed here are for example only: additional binary DEBs are available from the downloads site. If your architecture is not available, you can build binary DEB from the Debian DSC (see see Section 6.4.2 [Building from the Debian DSC], page 74).

Architecture Independent

strsock-dev_0.9.2.4-0_all.deb
: The '`strsock-dev`' package contains the device definitions necessary to run applications programs developed for OpenSS7 STREAMS Sockets.[14]

---

[11] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[12] Note that only release packages that contain kernel modules and that export versioned symbols will contain a '`streams-info`' subpackage.

[13] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[14] Note that not all release packages contain devices. Only packages that provide *STREAMS* character device drivers need devices, and then only when the '`specfs`' or '`devfsd`' is not being used.

strsock-doc_0.9.2.4-0_all.deb
: The '`strsock-doc`' package contains this manual in plain text, postscript, '`pdf`' and '`html`' forms, along with the meta-information from the '`strsock`' package. It also contains all of the manual pages necessary for developing OpenSS7 STREAMS Sockets applications and OpenSS7 STREAMS Sockets *STREAMS* modules or drivers.

strsock-init_0.9.2.4-0_all.deb
: The '`strsock-init`' package contains the `init` scripts and provides the postinst scripts necessary to create kernel module preloads and modules definitions for all kernel module '`core`' subpackages.

strsock-source_0.9.2.4-0_all.deb
: The '`strsock-source`' package contains the source code necessary for building the OpenSS7 STREAMS Sockets release. It includes the `autoconf(1)` configuration utilities necessary to create and distribute tarballs, rpms and deb/dscs.[15]

Architecture Dependent

strsock-devel_0.9.2.4-0_i386.deb
: The '`strsock-devel`' package contains library archives for static compilation, header files to develop OpenSS7 STREAMS Sockets modules and drivers. This also includes the header files and static libraries required to compile OpenSS7 STREAMS Sockets applications programs.

strsock-lib_0.9.2.4-0_i386.deb
: The '`strsock-lib`' package contains the run-time shared libraries necessary to run application programs and utilities developed for the '`strsock`' package.[16]

STREAMS-Dependent DEB

STREAMS-Dependent DEB are dependent upon the specific *STREAMS* package being used, either *Linux STREAMS* or *Linux Fast-STREAMS*. Packages dependent upon *Linux STREAMS* will have '`LiS`' in the package name. Packages dependent upon *Linux Fast-STREAMS* will have '`streams`' in the package name. Note that some *STREAMS*-Dependent DEB are also Kernel-Dependent DEB as described below.

One of the following *STREAMS*-Dependent packages is required for your architecture. If your architecture is not on the list, you can build binary DEB from the Debian DSC (see see Section 6.4.2 [Building from the Debian DSC], page 74).

strsock-LiS-util_0.9.2.4-0_i386.deb
: The '`strsock-LiS-util`' package provides administrative and configuration test utilities and commands associated with the OpenSS7 STREAMS Sockets package. Because this package must link a *STREAMS*-specific library, it is a *STREAMS*-Dependent package. Use the '`strsock-LiS-util`' package if you have *LiS* installed.

---

[15] Note that not all releases have source DEB packages. Release packages that do not contain kernel modules do not generate a source DEB package.

[16] Note that not all release packages contain shared libraries, and, therefore, not all release packages contain this package.

strsock-streams-util_0.9.2.4-0_i386.deb
: The '`strsock-streams-util`' package provides administrative and configuration test utilities and commands associated with the OpenSS7 STREAMS Sockets package. Because this package must link a *STREAMS*-specific library, it is a *STREAMS*-Dependent package. Use the '`strsock-streams-util`' package if you have streams installed.

Kernel-Dependent DEB

Kernel-Dependent DEB are dependent on specific Linux Kernel Binary DEB releases. Packages are provided for popular released *Debian* kernels. Packages dependent upon *Debian* or other kernel DEB will have the '`_kversion`' kernel package version in the package name.

One of the following Kernel-Dependent packages is required for your architecture and kernel version. If your architecture or kernel version is not on the list, you can build binary DEB from the source DEB (see see Section 6.4.2 [Building from the Debian DSC], page 74).[17]

strsock-core-2.4.20-28.7_0.9.2.4-0_i386.deb
: The '`strsock-core`' package contains the loadable kernel modules that depend only on the kernel. This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version '`2.4.20-28.7`'.[18]

strsock-info-2.4.20-28.7_0.9.2.4-0_i386.deb
: The '`strsock-info`' package[19] contains the module symbol version information for the '`core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loading the actual kernel modules (from the '`core`' subpackage above). This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version '`2.4.20-28.7`'.[20]

strsock-LiS-core-2.4.20-28.7_0.9.2.4-0_i386.deb
: The '`strsock-LiS-core`' package contains the kernel modules that provide the OpenSS7 STREAMS Sockets *STREAMS* modules and drivers. This package is heavily tied to the *STREAMS* package and kernel for which it was compiled. This particular package applies to '`LiS`' (*Linux STREAMS*) on kernel version '`2.4.20-28.7`'.[21]

strsock-streams-core-2.4.20-28.7_0.9.2.4-0_i386.deb
: The '`strsock-streams-core`' package contains the kernel modules that provide the OpenSS7 STREAMS Sockets *STREAMS* modules and drivers. This package is heavily tied to the *STREAMS* package and kernel for which it was com-

[17] Note that on *Mandrakelinux*, unlike other DEB kernel distributions, kernel packages for the ix86 architectures are always placed in i586 architecture packages regardless of the true processor architecture of the kernel package. '`configure`' detects this and builds the appropriate packages.

[18] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example. Note also that only release packages that contain kernel modules will contain a '`core`' subpackage.

[19] Note that only release packages that contain kernel modules and that export versioned symbols will contain a '`info`' subpackage. Also, this subpackage is only applicable to 2.4 series kernels and is not necessary and not built for 2.6 series kernels.

[20] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[21] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

piled. This particular package applies to '`streams`' (*Linux Fast-STREAMS*) on kernel version '`2.4.20-28.7`'.[22]

strsock-LiS-info-2.4.20-28.7_0.9.2.4-0_i386.deb
: The '`strsock-LiS-info`' package[23] contains the module symbol version information for the '`LiS-core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loaded the actual kernel modules (from the '`LiS-core`' subpackage above). This package is heavily tied to the *STREAMS* package and kernel for which it was compiled. This particular package applies to '`LiS`' (*Linux STREAMS*) on kernel version '`2.4.20-28.7`'.[24]

strsock-streams-info-2.4.20-28.7_0.9.2.4-0_i386.deb
: The '`strsock-streams-info`' package[25] contains the module symbol version information for the '`streams-core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loaded the actual kernel modules (from the '`streams-core`' subpackage above). This package is heavily tied to the *STREAMS* package and kernel for which it was compiled. This particular package applies to '`streams`' (*Linux Fast-STREAMS*) on kernel version '`2.4.20-28.7`'.[26]

Configuration and Installation

To configure, build and install the Debian DEB, See Section 6.3.2 [Configuring the Debian DEB], page 59.

6.2.5 Downloading the Source RPM

If you cannot obtain a binary RPM for your architecture, or would like to roll you own binary RPM, download the following source RPM.

strsock-0.9.2.4-1.src.rpm
: This is the source RPM for the package. From this source RPM it is possible to build binary RPM for any supported architecture and for any 2.4 or 2.6 kernel, for either *Linux STREAMS* or *Linux Fast-STREAMS*.

Configuration

To configure the source RPM, See Section 6.3.3 [Configuring the Source RPM], page 59.

6.2.6 Downloading the Debian DSC

If you cannot obtain a binary DEB for your architecture, or would like to roll your own DEB, download the following Debian DSC.

---

[22] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[23] Note that only release packages that contain kernel modules and that export versioned symbols will contain a '`LiS-info`' subpackage.

[24] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[25] Note that only release packages that contain kernel modules and that export versioned symbols will contain a '`streams-info`' subpackage.

[26] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

strsock_0.9.2.4-0.dsc
strsock_0.9.2.4-0.tar.gz
: This is the Debian DSC for the package. From this Debian DSC it is possible to build binary DEB for any supported architecture and for any 2.4 or 2.6 kernel, for either *Linux STREAMS* or *Linux Fast-STREAMS*.

Configuration

To configure the source RPM, See Section 6.3.4 [Configuring the Debian DSC], page 63.

6.2.7 Downloading the Tar Ball

For non-`rpm(1)` and non-`dpkg(1)` architectures, download the tarball as follows:

strsock-0.9.2.4.tar.gz
strsock-0.9.2.4.tar.bz2
: These are the `tar(1)` balls for the release. These `tar(1)` balls contain the `autoconf(1)` distribution which includes all the source necessary for building and installing the package. These tarballs will even build Source RPM and Binary RPM on `rpm(1)` architectures and Debian DSC and DEB on `dpkg(1)` architectures.

The tar ball may be downloaded easily with `wget(1)` as follows:

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.bz2
```

or

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.gz
```

Note that you will need an *OpenSS7 Project* user name and password to download release candidates (which are only available to subscribers and sponsors of the *OpenSS7 Project*).

Unpacking the Archive

After downloading one of the tar balls, unpack the archive using one of the following commands:

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.gz
% tar -xzvf strsock-0.9.2.4.tar.gz
```

or

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.bz2
% tar -xjvf strsock-0.9.2.4.tar.bz2
```

Either will create a subdirectory name ‘`strsock-0.9.2.4`’ containing all of the files and subdirectories for the `strsock` package.

Configuration

To configure and install the tar ball, See Section 6.3.5 [Configuring the Tar Ball], page 63.

6.2.8 Downloading from CVS

If you are a subscriber or sponsor of The OpenSS7 Project with CVS archive access privileges then you can download release, mid-release or release candidate versions of the '`strsock`' package from the project CVS archive.

The OpenSS7 STREAMS Sockets package is located in the '`strsock`' module of '`/var/cvs`'. For release tag information, see Chapter 5 [Releases], page 33.

To access the archive from the project CVS pserver, use the following commands to check out a version from the archive:

```
% export CVSROOT='-d:pserver:username@cvs.openss7.com:2401/var/cvs'
% cvs login
Password: *********
% cvs co -r strsock_0.9.2.4 strsock
% cvs logout
```

It is, of course, possible to check out by date or by other criteria. For more information, see `cvs(1)`.

Preparing the CVS Working Directory

Although public releases of the '`strsock`' package do not require reconfiguration, creating a configurable directory from the CVS archive requires tools not normally distributed with the other releases.

The build host requires the following GNU tools:

- `m4 1.4.12`
- `autoconf 2.63`
- `automake 1.10.1`
- `libtool 2.2.4`
- `gettext 0.17`
- `flex 2.5.33`
- `bison 2.3`

Most desktop development GNU/Linux distributions wil have these tools; however, some non-development or server-style installations might not and they must be installed separately.[27]

Also, these tools can be acquired from the FSF website in the free software directory, and also at the following locations:

- m4-1.4.12
- autoconf-2.63

[27] Older version of bison (2.0) and the older version of flex (2.5.4a) are also suitable. Where possible, use the more recent bison 2.3 and flex 2.5.33.

- automake-1.10.1
- libtool-2.2.4
- gettext-0.17
- flex-2.5.33
- bison-2.3

It should be stressed that, in particular, the `autoconf(1)`, and `automake(1)`, must be at version releases `2.63` and `1.10.1`. *The versions normally distributed in some mainstream GNU/Linux distributions are, in fact, much older than these versions.*[28] GNU version of these packages configured and installed to default directories will install in '`/usr/local/`' allowing them to coexist with distribution installed versions.

For building documentation, the build host also requires the following documentation tools:

- `gs 6.51` or `ghostscript 6.51`, or newer.
- `tetex 3.0` or `texlive 2007`, or newer.
- `texinfo 4.13a` or newer.
- `transfig 3.2.3d` or newer.
- `imagemagick 5.3.8` or `ImageMagick 5.3.8`, or newer.
- `groff 1.17.2` or newer.
- `gnuplot 3.7` or newer.
- `latex2html 1.62` or newer.

Most desktop GNU/Linux distributions will have these tools; however, some server-style installations (e.g. *Ubuntu*-server, *SLES 9* or *Fedora 6 or 7*) will not and they must be installed separately.[29]

Note that `texinfo 4.12` must not be used as it breaks the build process.

For uncooked manual pages, the entire `groff(1)` package is required on older *Debian* and *Ubuntu* systems (the base package did not include `grefer(1)` which is used extensively by uncooked manual pages). The following will get what you need on older systems:

```
Debian: % apt-get install groff_ext
Ubuntu: % apt-get install groff
```

On newer systems, simply:

```
% apt-get install groff
```

In addition, the build host requires a complete tool chain for compiling for the target host, including kernel tools such as `genksyms(8)` and others.

[28] A notable exception is Debian and Fedora 7. Note that on Fedora 7 the `gettext-devel` package must be installed.

[29] In particular, for *CentOS*, *Fedora 6 or 7*, the `tetex-latex` and `gnuplot` packages must be loaded as well. Note also that the `latex2html` used to be part of the `textex` package (or subpackages) but is now often packaged on its own. Recent distributions such as SUSE 11.0 and Fedora 9 use the `texlive` package instead of the `tetex` package.

If you wish to package '`rpms`' on an `rpm(1)` system, or '`debs`' on a `dpkg(1)` system, you will need the appropriate tool chain. Systems based on `rpm(1)` typically have the necessary tool chain available, however, `dpkg(1)` systems do not. The following on a *Debian* or *Ubuntu* system will get what you need:

```
% apt-get install debhelper
% apt-get install fakeroot
```

To generate a configuration script and the necessary scriptlets required by the GNU `autoconf(1)` system, execute the following commands on the working directory:

```
% autoreconf -fiv strsock
```

where, '`strsock`' is the name of the directory to where the working copy was checked out under the previous step. This command generates the '`configure`' script and other missing pieces that are normally distributed with the release Tar Balls, SRPMs and DSCs.

Make sure that '`autoreconf --version`' returns '`2.63`'. Otherwise, you may need to perform something like the following:

```
% PATH="/usr/local/bin:$PATH"
% autoreconf -fiv strsock
```

After reconfiguring the directory, the package can then be configured and built using the same instructions as are used for the Tar Ball, see Section 6.3.5 [Configuring the Tar Ball], page 63, and Section 6.4.3 [Building from the Tar Ball], page 74.

Do note, however, that `make(1)` will rebuild the documentation that is normally released with the package. Additional tools may be necessary for building the documentation. To avoid building and installing the documentation, use the '`--disable-devel`' or '`--disable-docs`' option to configure described in Section 6.3.5 [Configuring the Tar Ball], page 63.

When configuring the package in a working directory and while working a change-compile-test cycle that involves configuration macros or documentation, I find it of great advantage to invoke the GNU '`configure`' options '`--enable-maintainer-mode`', '`--enable-dependency-tracking`' and '`--disable-devel`'. The first of these three options will add maintainer-specific targets to any generated '`Makefile`', the second option will invoke automatic dependency tracking within the '`Makefile`' so rebuilds after changes to macro, source or documentation files will be automatically rebuilt; and the last option will suppress rebuilding and reinstalling documentation manual pages and header files. Header files will still be available under the '`/usr/src`' directory.

6.3 Configuration

6.3.1 Configuring the Binary RPM

In general the binary RPM do not require any configuration, however, during installation it is possible to relocate some of the installation directories. This allows some degree of customization. Relocations that are available on the binary RPM are as follows:

‘`strsock-LiS-core-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm`’
‘`strsock-streams-core-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm`’

‘`/lib/modules/2.4.20-28.7`’
: This relocatable directory contains the kernel modules that provide the strsock *STREAMS* core, drivers and modules.[30]

‘`strsock-LiS-info-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm`’
‘`strsock-streams-info-2.4.20-28.7-0.9.2.4-1.7.2.i686.rpm`’

‘`/usr/include/strsock/2.4.20-28.7`’
: This relocatable directory contains the kernel module exported symbol information that allows other kernel modules to be compiled against the correct version of the strsock package.[31]

‘`strsock-dev-0.9.2.4-1.7.2.i686.rpm`’
: (not relocatable)

‘`strsock-devel-0.9.2.4-1.7.2.i686.rpm`’

‘`/usr/lib`’
: This relocatable directory contains strsock libraries.

‘`/usr/include/strsock`’
: This relocatable directory contains strsock header files.

‘`strsock-doc-0.9.2.4-1.7.2.i686.rpm`’

‘`/usr/share/doc`’
: This relocatable directory contains all package specific documentation (including this manual). The subdirectory in this directory is the ‘`strsock-0.9.2.4`’ directory.

‘`/usr/share/info`’
: This relocatable directory contains info files (including the info version of this manual).

‘`/usr/share/man`’
: This relocatable directory contains manual pages.

‘`strsock-LiS-lib-0.9.2.4-1.7.2.i686.rpm`’
‘`strsock-streams-lib-0.9.2.4-1.7.2.i686.rpm`’

‘`/usr/lib`’
: This relocatable directory contains the run-time shared libraries necessary to run applications programs and utilities developed for OpenSS7 STREAMS Sockets.

---

[30] Note that the ‘`_kversion`’ of ‘`2.4.20-28.7`’ is only an example.

[31] Note that the ‘`_kversion`’ of ‘`2.4.20-28.7`’ is only an example. Also, note that the ‘`info`’ subpackage is only applicable to the 2.4 kernel series.

‘`/usr/share/locale`’
: This relocatable directory contains the locale information for shared library files.

‘`strsock-source-0.9.2.4-1.7.2.i686.rpm`’

‘`/usr/src`’
: This relocatable directory contains the source code.

‘`strsock-LiS-util-0.9.2.4-1.7.2.i686.rpm`’
‘`strsock-streams-util-0.9.2.4-1.7.2.i686.rpm`’

‘`/usr/bin`’
: This relocatable directory contains binary programs and utilities.

‘`/usr/sbin`’
: This relocatable directory contains system binary programs and utilities.

‘`/usr/libexec`’
: This relocatable directory contains test programs.

‘`/etc`’ This relocatable directory contains `init` scripts and configuration information.

Installation

To install the binary RPM, See Section 6.5.1 [Installing the Binary RPM], page 75.

6.3.2 Configuring the Debian DEB

In general the binary DEB do not require any configuration.

Installation

To install the Debian DEB, See Section 6.5.2 [Installing the Debian DEB], page 76.

6.3.3 Configuring the Source RPM

When building from the source RPM (see Section 6.4.1 [Building from the Source RPM], page 73), the rebuild process uses a number of macros from the user's ‘`.rpmmacros`’ file as described in `rpm(8)`.

Following is an example of the ‘`~/.rpmmacros`’ file that I use for rebuilding RPMS:

```
#
# RPM macros for building rpms
#

%vendor OpenSS7 Corporation
%distribution OpenSS7
%disturl http://www.openss7.org/
%packager Brian Bidulock <bidulock@openss7.org>
%url http://www.openss7.org/

%_signature gpg
%_gpg_path /home/brian/.gnupg
%_gpg_name openss7@openss7.org
%_gpgbin /usr/bin/gpg

%_source_payload w9.bzdio
%_binary_payload w9.bzdio

%_unpackaged_files_terminate_build 1
%_missing_doc_files_terminate_build 1
%_use_internal_dependency_generator 0
%_repackage_all_erasures 0
%_rollback_transaction_on_failure 0

%configure2_5x %configure
%make make
```

When building from the source RPM (see Section 6.4.1 [Building from the Source RPM], page 73), it is possible to pass a number of additional configuration options to the `rpmbuild(1)` process.

The additional configuration options are described below.

Note that distributions that use older versions of rpm do not have the '`--with`' or '`--without`' options defined. To achieve the same effect as:

```
--with someparm=somearg
```

do:

```
--define "_with_someparm --with-someparm=somearg"
```

This is a generic description of common `rpmbuild(1)` options. Not all `rpmbuild(1)` options are applicable to all SRPMs. Options that are kernel module specific are only applicable to SRPMs that build kernel modules. *STREAMS* options are only applicable to SRPMs that provide or require *STREAMS*.

`--define "_kversion $PACKAGE_KVERSION"`
: Specifies the kernel version other than the running kernel for which to build. If `_kversion` is not defined when rebuilding, the environment variable *PACKAGE_KVERSION* is used. If the environment variable *PACKAGE_KVERSION* is not defined, then the version of the running kernel (i.e. discovered with '`uname -r`') is used as the target version for kernel-dependent packages. This option can also be defined in an '`.rpmspec`' file using the macro name '`_kversion`'.

`--with checks`
`--without checks`
: Enable or disable preinstall checks. Each packages supports a number of preinstall checks that can be performed by invoking the '`check`' target with `automake(1)`. These currently consist of checking each kernel module for unresolved kernel symbols, checking for documentation for exported kernel module symbols, checking for documentation for exported library symbols, checking for standard options for build and installable programs, checking for documentation for built and installable programs. Normally these checks are only run in maintainer mode, but can be enabled and disabled with this option.

`--with k-optimize=HOW`
`--without k-optimize`
: Specify '`HOW`' optimization, *normal*, *size*, *speed* or *quick*. *size* compiles kernel modules `-Os`, *speed* compiles kernel modules `-O3`, and *quick* compiles kernel modules `-O0`. The default is *normal*. Use with care.

`--with cooked-manpages`
`--without cooked-manpages`
: Some systems do not like `grefer(1)` references in manual pages.[32] This option will cook `soelim(1)`, `refer(1)`, `tbl(1)` and `pic(1)` commands from the manual pages and also strip `groff(1)` comments. The default is to leave manual pages uncooked: they are actually smaller that way.

`--with public`
`--without public`
: Release public packages or private packages. This option has no effect on the '`strsock`' package. The default is to release public packages.

`--with k-debug`
`--without k-debug`
: Specifies whether kernel debugging is to be performed on the build kernel modules. Mutually exclusive with `test` and `safe` below. This has the effect of removing static and inline attributes from functions and invoking all debugging macros in the code. The default is to not perform kernel debugging.

`--with k-test`
`--without k-test`
: Specifies whether kernel testing is to be performed. Mutually exclusive with `debug` above and `safe` below. This has the effect of removing static and inline attributes from functions and invoking most debugging macros in the code. The default is to not perform kernel testing.

[32] In particular, some *Debian* systems do not load the `groff(1)` extensions package and do not have `grefer(1)` installed. Although this is an oversight on the configuration of the particular *Debian* system, we accomodate such misconfiguration with this feature.

`--with k-safe`
`--without k-safe`
: Specifies whether kernel saftey is to be performed. Mutually exclusive with `debug` and `test` above. This has the effect of invoking some more pedantic assertion macros in the code. The default is not to apply kernel safety.

`--with k-inline`
`--without k-inline`
: Specifies whether kernel `inline` functions are to be placed inline. This has the effect of adding the '`-finline-functions`' flag to *CFLAGS* for compiling kernel modules. Linux 2.4 kernels are normally compiled '`-O2`' which does not respect the `inline` directive. This compiles kernel modules with '`-finline-functions`' to get closer to '`-O3`' optimization. For better optimization controls, See Section 6.3.5 [Configuring the Tar Ball], page 63.

`--with k-modversions`
`--without k-modversions`
: Specifies whether kernel symbol versions are to be applied to symbols exported by package kernel modules. The default is to version exported module symbols. This package does not export symbols so this option has no effect.

`--with devfs`
`--without devfs`
: Specifies whether the build is for a device file system daemon enabled system with autoloading, or not. The default is to build for `devfsd(1)` autoloading when *CONFIG_DEVFS_FS* is defined in the target kernel. The '`rebuild`' target uses this option to signal to the RPM spec file that the '`dev`' subpackage need not be built. This option does not appear when the package has no devices.

`--with devel`
`--without devel`
: Specifies whether to build development environment packages such as those that include header files, static libraries, manual pages and `texinfo(1)` documentation. The default is to build development environment packages. This option can be useful when building for an embedded target where only the runtime components are desired.

`--with docs`
`--without docs`
: Specifies whether to build and install major documentation such manual pages and `texinfo(1)` documentation. The default is to build and install documentation. This option can be useful when building for an embedded target where only the runtime and static compile components are desired, but not major documentation. This option does not override the setting of `--without devel`.

`--with tools`
`--without tools`
: Specifies whether user space packages are to be built. The default is to build user space packages. This option can be useful when rebuilding for multiple architectures and target kernels. The '`rebuild`' `automake(1)` target uses this

feature when rebuilding for all available architectures and kernels, to rebuild user packages once per architecture instead of once per kernel.

`--with modules`
`--without modules`
: Specifies whether kernel modules packages are to be built. The default is to build kernel module packages. This option can be useful when rebuilding for multiple architectures and target kernels. The '`rebuild`' `automake(1)` target uses this feature to rebuild for all available architectures and kernels.

`--with lis`
`--without lis`
: Specifies that the package is to be rebuilt against *Linux STREAMS*. The default is to automatically identify whether '`LiS`' or '`streams`' is loaded on the build system and build accordingly.

`--with lfs`
`--without lfs`
: Specifies that the package is to be rebuilt against *Linux Fast-STREAMS*. The default is to automatically identify whether '`LiS`' or '`streams`' is loaded on the build system and build accordingly.

In general, the default values of these options are sufficient for most purposes and no options need be provided when rebuilding the Source RPMs.

Build

To build from the source RPM, See Section 6.4.1 [Building from the Source RPM], page 73.

6.3.4 Configuring the Debian DSC

The Debian DSC can be configured by passing options in the environment variable *BUILD_DEBOPTIONS*. The options placed in this variable take the same form as those passed to the '`configure`' script, See Section 6.3.5 [Configuring the Tar Ball], page 63. For an example, See Section 6.4.2 [Building from the Debian DSC], page 74.

Build

To build from the Debian DSC, See Section 6.4.2 [Building from the Debian DSC], page 74.

6.3.5 Configuring the Tar Ball

All of the normal GNU `autoconf(1)` configuration options and environment variables apply. Additional options and environment variables are provided to tailor or customize the build and are described below.

6.3.5.1 Configure Options

This is a generic description of common '`configure`' options that are in addition to those provided by `autoconf(1)`, `automake(1)`, `libtool(1)` and `gettext(1)`.

Not all '`configure`' options are applicable to all release packages. Options that are kernel module specific are only applicable to release packages that build kernel modules. *STREAMS* options are only applicable to release packages that provide or require *STREAMS*.

Following are the additional '`configure`' options, their meaning and use:

`--enable-checks`
`--disable-checks`
: Enable or disable preinstall checks. Each release package supports a number of preinstall checks that can be performed by invoking the '`check`' target with `make(1)`. These currently consist of checking each kernel module for unresolved kernel symbols, checking for documentation for exported kernel module symbols, checking for documentation for exported library symbols, checking for standard options for build and installable programs, checking for documentation for built and installable programs. Normally these checks are only run in maintainer mode, but can be enabled and disabled with this option.

`--enable-autotest`
`--disable-autotest`
: Enable or disable pre- and post-installation testing. Each release package supports a number of `autotest` test suites that can be performed by invoking the '`installcheck`' target with `make(1)`. These currently consist of running installed modules, commands and binaries against a number of specific test cases. Normally these checks are only run in maintainer mode, but can be enabled and disabled with this option.

`--disable-compress-manpages`
: Compress manual pages with '`gzip -9`' or '`bzip2 -9`' or leave them uncompressed. The default is to compress manual pages with '`gzip -9`' or '`bzip2 -9`' if a single compressed manual page exists in the target installation directory ('`--mandir`'). This disables automatic compression.

`--disable-public`
: Disable public release. This option is not usable on public releases and only has a usable effect on OpenSS7 STREAMS Sockets when the package is acquired from CVS. In particular, the *STREAMS SS7/VoIP/ISDN/SIGTRAN Stacks* (`strss7-0.9a.8`) release package has a large number of non-public components. Specifying this option will cause the package to build and install all private release components in addition to the public release components. This option affects all release packages. Most release packages do not have private release components.

`--disable-initscripts`
: Disables the installation of `init` scripts. The default is to configure and install `init` scripts and their associated configuration files.

  Although the default is to install `init` scripts, installation attempts to detect a System V `init` script configuration, and if one is not found, the `init` scripts are installed into the appropriate directories, but the symbolic links to the run level script directories are not generated and the script is not invoked. Therefore, it is safe to leave this option unchanged, even on distributions that do not support System V `init` script layout.

`--disable-32bit-libs`
: Disables the build and install of 32-bit compatibility libraries and test binaries on 64-bit systems that support 32-bit compatibility. The default is to build

and install 32-bit compatibility libraries and test binaries. This option can be usefule when configuring for an embedded target where only native shared libraries and binaries are desired.

`--disable-devel`
: Disables the installation of development environment components such as header files, static libraries, manual pages and `texinfo(1)` documentation. The default is to install development environment components. This option can be useful when configuring for an embedded target where only the runtime components are desired, or when performing a edit-compile-test cycle.

`--disable-docs`
: Disables the build and installation of major documentation such manual pages and `texinfo(1)` documentation. The default is to build and install documentation. This option can be useful when building for an embedded target where only the runtime and static compile components are desired, but not major documentation. This option does not override the setting of '`--disable-devel`'.

`--enable-tools`
: Specifies whether user space programs and libraries are to be built and installed. The default is to build and install user space programs and libraries. This option can be useful when rebuilding for multiple architectures and target kernels, particularly under `rpm(1)` or `dpkg(1)`. The '`rebuild`' `automake(1)` target uses this feature when rebuilding RPMs for all available architectures and kernels, to rebuild user packages once per architecture instead of once per kernel.

`--enable-modules`
: Specifies whether kernel modules are to be built and installed. The default is to build and install kernel modules. This option can be useful when rebuilding for multiple architectures and target kernels, particularly under `rpm(1)` or `dpkg(1)`. The '`rebuild`' `automake(1)` target uses this feature to rebuild for all available architectures and kernels. This option has no effect for release packages that do not provide kernel modules.

`--enable-arch`
: Specifies whether architectural dependent package components are to be built and installed. This option can be useful when rebuilding for multiple architectures and target kernels, particularly under `dpkg(1)`. The default is to configure, build and install architecture dependent package components. This option has no effect for release packages that do not provide architecture dependent components.

`--enable-indep`
: Specifies whether architecture independent package components are to be built and installed. This option can be useful when rebuilding for multiple architectures and target kernels, particularly under `dpkg(1)`. The default is to configure, build and install architecture independent package components. This options has no effect for release packages that do not provide architecture independent components.

`--enable-k-inline`
: Enable kernel inline functions. Most Linux kernels build without '`-finline-functions`'. This option adds the '`-finline-functions`' and '`-Winline`' flags to the compilation of kernel modules. Use with care. This option has no effect for release packages that do not provide kernel modules.

`--enable-k-safe`
: Enable kernel module run-time safety checks. Specifies whether kernel safety is to be performed. This option is mutually exclusive with '`--enable-k-test`' and '`--enable-k-debug`' below. This has the effect of invoking some more pedantic assertion macros in the code. The default is not to apply kernel safety. This option has no effect for release packages that have are no kernel modules.

`--enable-k-test`
: Enable kernel module run-time testing. Specifies whether kernel testing is to be performed. This option is mutually exclusive with '`--enable-k-safe`' above and '`--enable-k-debug`' below. This has the effect of remove `static` and `inline` attributes from functions and invoking most non-performance affecting debugging macros in the code. The default is not to perform kernel testing. This option has no effect for release packages that do not provide kernel modules.

`--enable-k-debug`
: Enable kernel module run-time debugging. Specifies whether kernel debugging is to be performed. This option is mutually exclusive with '`--enable-k-safe`' and '`--enable-k-test`' above. This has the effect of removing `static` and `inline` attributes from functions and invoking all debugging macros in the code (including performance-affecting debug macros). The default is to not perform kernel debugging. This option has no effect for release packages that do not provide kernel modules.

`--disable-k-modversions`
: Disable module versions on `strsock` symbols. Specifies whether kernel symbol versions are to be used on symbols exported from built `strsock` modules. The default is to provide kernel symbol versions on all exported symbols. This option has no effect for release packages that do not provide kernel modules.

`--enable-devfs`
`--disable-devfs`
: Specifies whether the build is for a device file system daemon enabled system with autoloading, or not. The default is to build for `devfsd(8)` autoloading when *CONFIG_DEVFS_FS* is defined in the target kernel. The '`reuild`' `automake(1)` target uses this option to signal to the RPM spec file that the '`dev`' subpackage need not be built. This option has no effect for release packages that do not provide devices.

`--with-gpg-user=GNUPGUSER`
: Specify the `gpg(1)` '`GNUPGUSER`' for signing RPMs and tarballs. The default is the content of the environment variable *GNUPGUSER*. If unspecified, the `gpg(1)` program will normally use the user name of the account invoking the `gpg(1)` program. For building source RPMs, the RPM macro '`_gpg_name`' will override this setting.

`--with-gpg-home=GNUPGHOME`
: Specify the '`GNUPGHOME`' directory for signing RPMs and tarballs. The default is the user's '`~/.gpg`' directory. For building source RPMs, the RPM macro '`_gpg_path`' will override this setting.

`--with-pkg-epoch=EPOCH`
: Specifies the epoch for the package. This is neither used for `rpm(1)` nor `dpkg(1)` packages, it applies to the tarball release as a whole. The default is the contents of the '`.pkgepoch`' file in the release package source directory or, if that file does not exist, zero (0).

`--with-pkg-release=RELEASE`
: Specifies the release for the package. This is neither used for `rpm(1)` nor `dpkg(1)` packages, it applies to the tarball release as a whole. The default is the contents of the '`.pkgrelease`' file in the release package source directory or, if that file does not exist, one (1). This is the number after the last point in the package version number.

`--with-pkg-distdir=DIR`
: Specifies the distribution directory for the package. This is used by the maintainer for building distributions of tarballs. This is the directory into which archives are copied for distribution. The default is the top build directory.

`--with-cooked-manpages`
: Convert manual pages to remove macro dependencies and `grefer(1)` references. Some systems do not like `grefer(1)` references in manual pages.[33] This option will cook `soelim(1)`, `refer(1)`, `tbl(1)` and `pic(1)` commands from the manual pages and also strip `groff(1)` comments. The default is to leave manual pages uncooked (they are actually smaller that way).

`--with-rpm-epoch=PACKAGE_EPOCH`
: Specify the '`PACKAGE_EPOCH`' for the RPM spec file. The default is to use the RPM epoch contained in the release package file '`.rpmepoch`'.

`--with-rpm-release=PACKAGE_RPMRELEASE`
: Specify the '`PACKAGE_RPMRELEASE`' for the RPM '`spec`' file. The default is to use the RPM release contained in the release package file '`.rpmrelease`'.

`--with-rpm-extra=PACKAGE_RPMEXTRA`
: Specify the '`PACKAGE_RPMEXTRA`' extra release information for the RPM spec file. The default is to use the RPM extra release information contained in the release package file '`.rpmextra`'. Otherwise, this value will be determined from automatic detection of the RPM distribution.

`--with-rpm-topdir=PACKAGE_RPMTOPDIR`
: Specify the '`PACKAGE_RPMTOPDIR`' top directory for RPMs. If specified with a null '`PACKAGE_RPMTOPDIR`', the default directory for the RPM distribution will be used. If this option is not provided on the command line, the top build directory will be used as the RPM top directory as well.

[33] In particular, some *Debian* or *Ubuntu* systems do not load the `groff(1)` extensions package and do not have `grefer(1)` installed. Although this is an oversight on the configuration of the particular *Debian* or *Ubuntu* system, we accomodate such misconfiguration with this feature.

`--with-deb-epoch=EPOCH`
: Specify the '`PACKAGE_DEBEPOCH`' for the DEB control file. The default is to use the DEB epoch contained in the release package file '`.debepoch`'.

`--with-deb-release=RELEASE`
: Specify the '`PACKAGE_DEBRELEASE`' for the DEB control file. The default is to use the DEB release contained in the release package file '`.debrelease`'.

`--with-deb-topdir=DIR`
: Specify the '`PACKAGE_DEBTOPDIR`' top directory for DEBs. If specified with a null '`PACKAGE_DEBTOPDIR`', the default directory for the DEB distribution will be used. If this option is not provided on the command line, the top build directory will be used as the DEB top directory as well.

`--with-k-release=PACKAGE_KRELEASE`
: Specify the '`PACKAGE_KRELEASE`' release of the Linux kernel for which the build is targeted. When not cross compiling, if this option is not set, the build will be targeted at the kernel running in the build environment (e.g., '`uname -r`'). When cross-compiling this option must be specified or the configure script will generate an error and terminate.

`--with-k-linkage=PACKAGE_KLINKAGE`
: Specify the '`PACKAGE_KLINKAGE`' for kernel module linkage. This can be one of the following:

  - '`loadable`' – loadable kernel modules
  - '`linkable`' – linkable kernel objects

  The default is to build loadable kernel modules.

`--with-k-modules=K-MODULES-DIR`
: Specify the '`K-MODULES-DIR`' directory to which kernel modules will be installed. The default is based on the option '`--with-k-release`', '`--with-k-prefix`' and '`--with-k-rootdir`'. The default is '`DESTDIR`'/'`K-MODULES-DIR`' which is typically '*`DESTDIR`*`/lib/modules/`*`PACKAGE_KRELEASE`*`/`'. This directory is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message.

`--with-k-build=K-BUILD-DIR`
: Specify the '`K-BUILD-DIR`' base kernel build directory in which configured kernel source resides. The default is '*`DESTDIR`*`/`*`K-MODULES-DIR`*`/build`'. This directory is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message.

`--with-k-source=K-SOURCE-DIR`
: Specify the '`K-SOURCE-DIR`' base kernel build directory in which configured kernel source resides. The default is '*`DESTDIR`*`/`*`K-MODULES-DIR`*`/source`'. This directory is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message.

`--with-k-modver=K-MODVER-FILE`
: Specify the '`K-MODVER-FILE`' kernel module versions file. The default is '*K-BUILD-DIR*`/Module.symvers`'. This file is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message.

`--with-k-sysmap=K-SYSMAP-FILE`
: Specify the '`K-SYSMAP-FILE`' kernel system map file. The default is '*K-BUILD-DIR*`/System.map`'. This file is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message.

`--with-k-archdir=K-ARCHDIR`
: Specify the '`K-ARCHDIR`' kernel source architecture specific directory. The default is '*DESTDIR/K-SOURCE-DIR*`/arch`'. This directory is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message.

`--with-k-machdir=K-MACHDIR`
: Specify the '`K-MACHDIR`' kernel source machine specific directory. The default is '*DESTDIR/K-SOURCE-DIR/target_cpu*'. This directory is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message.

`--with-k-config=K-CONFIG`
: Specify the '`K-CONFIG`' kernel configuration file. The default is '*BOOT*`/config-`*K-RELEASE*'. This configuration file is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message.

`--with-k-optimize=HOW`
`--without-k-optimize`
: Specify '`HOW`' optimization, *normal*, *size*, *speed* or *quick*. *size* compiles kernel modules `-Os`, *speed* compiles kernel modules `-O3`, and *quick* compiles kernel modules `-O0`. The default is *normal*. Use with care. The most common use of this option is to specify '`--with-k-optimize=speed --disable-k-safe`' to compile for maximum performance. Nevertheless, even these setting are *ricing* and the resulting kernel modules will only be about 5% faster.

`--with-lis[=LIS-DIR]`
`--without-lis`
: Specify the '`LIS-DIR`' directory in which to find *LiS* headers. Also specifies that the build is to be made against Linux STREAMS. The default is '`/usr/include/LiS`' if it exists, '`no`' otherwise. This directory is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message. This option has no effect on release packages that do not use the *STREAMS* subsystem.

`--with-lfs[=LFS-DIR]`
`--without-lfs`
: Specify the '`LFS-DIR`' directory in which to find *LfS* headers. Also specifies that the build is to be made against Linux Fast-STREAMS. The default is '`/usr/include/streams`' if it exists, '`no`' otherwise. This directory is normally located by the '`configure`' script and need only be provided for special cross-build environments or when requested by a '`configure`' script error message. This option has no effect on release packages that do not use the *STREAMS* subsystem.

`--with-strconf-master=STRCONF_CONFIG`
: Specify the '`STRCONF_CONFIG`' file name to which the configuration master file is written. The default is '`Config.master`'. This option has no effect on release packages that do not use the *STREAMS* subsystem and the strconf scripts. This option should not be specified when configuring the master package as the setting for all add-on packages will conflict.

`--with-base-major=STRCONF_MAJBASE`
: Start numbering for major devices at '`STRCONF_MAJBASE`'. The default is '`230`'. This option has no effect on release packages that do not use the *STREAMS* subsystem and the strconf scripts. This option should not be specified when configuring the master package as the setting for all add-on packages will conflict.

6.3.5.2 Environment Variables

Following are additional environment variables to '`configure`', their meaning and use:

GPG
: GPG signature command. This is used for signing distributions by the maintainer. By default, '`configure`' will search for this tool.

GNUPGUSER
: GPG user name. This is used for signing distributions by the maintainer.

GNUPGHOME
: GPG home directory. This is used for signing distributions by the maintainer.

GPGPASSWD
: GPG password for signing. This is used for signing distributions by the maintainer. This environment variable is not maintained by the '`configure`' script and should only be used on an isolated system.

SOELIM
: Roff source elimination command, `soelim(1)`. This is only necessary when the option '`--with-cooked-manpages`' has been specified and '`configure`' cannot find the proper `soelim(1)` command. By default, '`configure`' will search for this tool.

REFER
: Roff references command, `refer(1)`. This is only necessary when the option '`--with-cooked-manpages`' has been specified and '`configure`' cannot find the proper `refer(1)` command. By default, '`configure`' will search for this tool.

TBL
: Roff table command, `tbl(1)`. This is only necessary when the option '`--with-cooked-manpages`' has been specified and '`configure`' cannot find the proper `tbl(1)` command. By default, '`configure`' will search for this tool.

PIC Roff picture command, `pic(1)`. This is only necessary when the option '`--with-cooked-manpages`' has been specified and '`configure`' cannot find the proper `pic(1)` command. By default, '`configure`' will search for this tool.

GZIP Default compression options provided to `GZIP_CMD`.

GZIP_CMD
Manpages (and kernel modules) compression commands, `gzip(1)`. This is only necessary when the option '`--without-compressed-manpages`' has *not* been specified and '`configure`' cannot find the proper `gzip(1)` command. By default, '`configure`' will search for this tool.

BZIP2 Default compression options provided to `BZIP2_CMD`

BZIP2_CMD
Manpages compression commands, `bzip2(1)`. This is only necessary when the option '`--without-compressed-manpages`' has *not* been specified and '`configure`' cannot find the proper `bzip2(1)` command. By default, '`configure`' will search for this tool.

MAKEWHATIS
Manpages apropros database rebuild command, `makewhatis(8)`. By default, '`configure`' will search for this tool. By default, '`configure`' will search for this tool.

CHKCONFIG
Chkconfig command, `chkconfig(8)`. This was used for installation of `init` scripts. All packages now come with `init_install(8)` and `init_remove(8)` scripts used to install and remove `init` scripts on both RPM and Debian systems.

RPM Rpm command, `rpm(1)`. This is only necessary for RPM builds. By default, '`configure`' will search for this tool.

RPMBUILD
Build RPM command, `rpmbuild(1)`. This is only necessary for RPM builds. By default, '`configure`' will search for this tool. `rpm(1)` will be used instead of `rpmbuild(1)` only if `rpmbuild(1)` cannot be found.

DPKG Dpkg comand, `dpkg(1)`. This command is used for building Debian packages. By default, '`configure`' will search for this tool.

DPKG_SOURCE
Dpkg-source command, `dpkg-source(1)`. This command is used for building Debian dsc packages. By default, '`configure`' will search for this tool.

DPKG_BUILDPACKAGE
Dpkg-buildpackage command, `dpkg-buildpackage(1)`. This command is used for building Debian deb packages. By default, '`configure`' will search for this tool.

DEB_BUILD_ARCH
Debian build architecture. This variable is used for building Debian packages. The default is the autoconf build architecture.

DEB_BUILD_GNU_CPU
: Debian build cpu. This variable is used for building Debian packages. The default is the autoconf build cpu.

DEB_BUILD_GNU_SYSTEM
: Debian build os. This variable is used for building Debian packages. The default is the autoconf build os.

DEB_BUILD_GNU_TYPE
: Debian build alias. This variable is used for building Debian packages. The default is the autoconf build alias.

DEB_HOST_ARCH
: Debian host architecture. This variable is used for building Debian packages. The default is the autoconf host architecture.

DEB_HOST_GNU_CPU
: Debian host cpu. This variable is used for building Debian packages. The default is the autoconf host cpu.

DEB_HOST_GNU_SYSTEM
: Debian host os. This variable is used for building Debian packages. The default is the autoconf host os.

DEB_HOST_GNU_TYPE
: Debian host alias. This variable is used for building Debian packages. The default is the autoconf host alias.

LDCONFIG
: Configure loader command, `ldconfig(8)`. Command used to configure the loader when libraries are installed. By default, '`configure`' will search for this tool.

DESTDIR
: Cross build root directory. Specifies the root directory for build and installation.

DEPMOD
: Build kernel module dependencies command, `depmod(8)`. This is used during installation of kernel modules to a running kernel to rebuild the modules dependency database. By default, '`configure`' will search for this tool.

MODPROBE
: Probe kernel module dependencies command, `modprobe(8)`. This is used during installation of kernel modules to a running kernel to remove old modules. By default, '`configure`' will search for this tool.

LSMOD
: List kernel modules command, `lsmod(8)`. This is used during installation of kernel modules to a running kernel to detect old modules for removal. By default, '`configure`' will search for this tool.

LSOF
: List open files command, `lsof(1)`. This is used during installation of kernel modules to a running kernel to detect old modules for removal. Processes owning the old kernel modules will be killed and the module removed. If the process restarts, the new module will be demand loaded. By default, '`configure`' will search for this tool.

GENKSYMS
: Generate kernel symbols command, `genksyms(8)`. This is used for generating module symbol versions during build. By default, '`configure`' will search for this tool.

KGENKSYMS
: Linux 2.6 generate kernel symbols command, `genksyms(8)`. This is used for generating module symbol version during build. By default, '`configure`' will search for this tool.

OBJDUMP
: Object dumping command, `objdump(1)`. This is used for listing information about object files. By default, '`configure`' will search for this tool.

NM
: Object symbol listing command, `nm(1)`. This is used for listing information about object files. By default, '`configure`' will search for this tool.

MODPOST_CACHE
: Cache file for `modpost(1)`. The version of the `modpost.sh` script that ships with each package can cache information to a cache file to speed multiple builds. This environment variable is used to specify a cache file.

AUTOM4TE
: Autom4te command, `autom4te(1)`. This is the executable used by `autotest` for pre- and post-installation checks. By default, '`configure`' will search for this tool.

AUTOTEST
: Autotest macro build command, `autom4te(1)`. This is the executable used by `autotest` for pre- and post-installation checks. By default, '`configure`' will search for this tool.

6.3.5.3 Build

To build from the tar ball, See Section 6.4.3 [Building from the Tar Ball], page 74.

6.4 Building

6.4.1 Building from the Source RPM

If you have downloaded the necessary source RPM (see Section 6.2.5 [Downloading the Source RPM], page 53), then the following instructions will rebuild the binary RPMs on your system. Once the binary RPMs are rebuilt, you may install them as described above (see Section 6.5.1 [Installing the Binary RPM], page 75).

The source RPM is rebuilt to binary RPMs as follows:

```
% wget http://www.openss7.org/rpms/SRPMS/strsock-0.9.2.4-1.src.rpm
% rpmbuild --rebuild -vv strsock-0.9.2.4-1.src.rpm
```

The rebuild process can also recognize a number of options that can be used to tweak the resulting binaries, See Section 6.3.3 [Configuring the Source RPM], page 59. These options are provided on the `rpm(1)` command line. For example:

```
% rpmbuild --rebuild -vv --target athlon-redhat-linux \
  --define "_kversion 2.4.20-28.7" \
  --with lfs -- strsock-0.9.2.4-1.src.rpm
```

will rebuild binary RPM for the '`2.4.20-28.7`' kernel for the '`athlon`' architecture against the *Linux Fast-STREAMS STREAMS* package.[34]

Installation

To install the resulting binary RPM, See Section 6.5.1 [Installing the Binary RPM], page 75.

6.4.2 Building from the Debian DSC

If you have downloaded the necessary Debian DSC (see Section 6.2.6 [Downloading the Debian DSC], page 53), then the following instructions will rebuild the binary DEBs on your system. Once the binary DEBs are rebuilt, you may install them as described above (see Section 6.5.2 [Installing the Debian DEB], page 76).

The Debian DSC is rebuilt to binary DEBs as follows:

```
% wget http://www.openss7.org/debian/strsock_0.9.2.4-0.dsc
% wget http://www.openss7.org/debian/strsock_0.9.2.4-0.tar.gz
% dpkg-buildpackage -v strsock_0.9.2.4-0.dsc
```

The rebuild process can also recognize a number of options that can be used to tweak the resulting binaries, See Section 6.3.4 [Configuring the Debian DSC], page 63. These options are provided in the environment variable *BUILD_DPKGOPTIONS* and have the same form as the options to '`configure`', See Section 6.3.5 [Configuring the Tar Ball], page 63. For example:

```
% BUILD_DEBOPTIONS='
        --with-lfs
        --with-k-release=2.4.20-28.7
        --host=athlon-debian-linux-gnu'
  dpkg-buildpackage -v \
  strsock_0.9.2.4-0.dsc
```

will rebuild binary DEB for the '`2.4.20-28.7`' kernel for the '`athlon`' architecture against the *Linux Fast-STREAMS STREAMS* package.[35]

Installation

To install the resulting binary DEB, See Section 6.5.2 [Installing the Debian DEB], page 76.

6.4.3 Building from the Tar Ball

If you have downloaded the tar ball (see Section 6.2.7 [Downloading the Tar Ball], page 54), then the following instructions will rebuild the package on your system. (Note that the build process does not required `root` privilege.)

[34] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

[35] Note that the '`_kversion`' of '`2.4.20-28.7`' is only an example.

6.4.3.1 Native Build

Following is an example of a native build against the running kernel:

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.bz2
% tar -xjvf strsock-0.9.2.4.tar.bz2
% pushd strsock-0.9.2.4
% ./configure
% make
% popd
```

6.4.3.2 Cross-Build

Following is an example for a cross-build. The kernel release version must always be specified for a cross-build.[36] If you are cross-building, specify the root for the build with environment variable *DESTDIR*. The cross-compile host must also be specified if different from the build host. Either the compiler and other tools must be in the usual places where GNU `autoconf(1)` can find them, or they must be specified with declarations such as '`CC=/usr/lib/ppc-linux/gcc`' on the '`configure`' command line.

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.bz2
% tar -xjvf strsock-0.9.2.4.tar.bz2
% pushd strsock-0.9.2.4
% ./configure DESTDIR="/some/other/root" \
--with-k-release=2.4.18 --host sparc-linux
% make
% popd
```

6.5 Installing

6.5.1 Installing the Binary RPM

If you have downloaded the necessary binary RPMs (see Section 6.2.3 [Downloading the Binary RPM], page 47), or have rebuilt binary RPMs using the source RPM (see Section 6.4.1 [Building from the Source RPM], page 73), then the following instructions will install the RPMs on your system. For additional information on `rpm(1)`, see `rpm(8)`.

```
% pushd RPMS/i686
% rpm -ihv strsock-*-0.9.2.4-1.7.2.i686.rpm
```

You must have the correct binary RPMs downloaded or built for this to be successful.

Some of the packages are relocatable and can have final installation directories altered with the '`--relocate`' option to `rpm(1)`, see `rpm(8)`. For example, the following will relocate the documentation and info directories:

[36] Because it *is* a cross-build, the kernel version on the build machine is unlikely to be the kernel version of the target machine, except by coincidence.

```
% pushd RPMS/i686
% rpm -ihv \
        --relocate '/usr/share/doc=/usr/local/share/doc' \
        --relocate '/usr/share/info=/usr/local/share/info' \
        -- strsock-doc-0.9.2.4-1.7.2.i686.rpm
```

The previous example will install the '`strsock-doc`' package by will relocate the documentation an info directory contents to the '`/usr/local`' version.

6.5.2 Installing the Debian DEB

If you have downloaded the necessary Debian DEBs (see Section 6.2.4 [Downloading the Debian DEB], page 50), or have rebuild binary DEBs using the Debian DSC (see Section 6.4.2 [Building from the Debian DSC], page 74), then the following instructions will install the DEBs on your system. For additional information see `dpkg(8)`.

```
% pushd debian
% dpkg -iv strsock-*_0.9.2.4-0_*.deb
```

You must have the correct '`.deb`' files downloaded or build for this to be successful.

6.5.3 Installing the Tar Ball

After the build process (see Section 6.4.3 [Building from the Tar Ball], page 74), installation only requires execution of one of two `automake(1)` targets:

'`make install`'
: The '`install`' `automake(1)` target will install all the components of the package. Root privilege is required to successfully invoke this target.

'`make install-strip`'
: The '`install-strip`' `automake(1)` target will install all the components of the package, but will strip unnecessary information out of the objects and compress manual pages. Root privilege is required to successfully invoke this target.

6.6 Removing

6.6.1 Removing the Binary RPM

To remove an installed version of the binary RPMs (whether obtained from the OpenSS7 binary RPM releases, or whether created by the source RPM), execute the following command:

```
% rpm -evv `rpm -qa | grep '^strsock-'`
```

For more information see `rpm(1)`.

6.6.2 Removing the Debian DEB

To remove and installed version of the Debian DEB (whether obtained from the OpenSS7 binary DEB releases, or whether created by the Debian DSC), execute the following command:

```
% dpkg -ev `dpkg -l | grep '^strsock-'`
```

For more information see dpkg(8).

6.6.3 Removing the Source RPM

To remove all the installed binary RPM build from the source RPM, see Section 6.6.1 [Removing the Binary RPM], page 76. Then simply remove the binary RPM package files and source RPM file. A command such as:

```
% find / -name 'strsock-*.rpm' -type f -print0 | xargs --null rm -f
```

should remove all 'strsock' RPMs from your system.

6.6.4 Removing the Debian DSC

To remove all the installed binary DEB build from the Debian DSC, see Section 6.6.2 [Removing the Debian DEB], page 76. Then simply remove the binary DEB package files and Debian DSC file. A command such as:

```
% find / \( -name 'strsock-*.deb' \
         -o -name 'strsock-*.dsc' \
         -o -name 'strsock-*.tar.* \
         \) -type f -print0 | xargs --null rm -f
```

should remove all 'strsock' DEBs, DSCs and TARs from your system.

6.6.5 Removing the Tar Ball

To remove a version installed from tar ball, change to the build directory where the package was built and use the 'uninstall' automake(1) target as follows:

```
% cd /usr/src/strsock
% make uninstall
% cd ..
% rm -fr strsock-0.9.2.4
% rm -f strsock-0.9.2.4.tar.gz
% rm -f strsock-0.9.2.4.tar.bz2
```

If you have inadvertently removed the build directory and, therefore, no longer have a configured directory from which to execute 'make uninstall', then perform all of the steps for configuration and installation (see Section 6.5.3 [Installing the Tar Ball], page 76) except the final installation and then perform the steps above.

6.7 Loading

6.7.1 Normal Module Loading

When OpenSS7 STREAMS Sockets installs, modules and drivers belonging to release packages are normally configured for demand loading. The 'install' and 'install-strip'

automake(1) targets will make the necessary changes to the '/etc/modules.conf' file and place the modules in an appropriate place in '/lib/modules/2.4.20-28.7/strsock'. The 'make install' process should have copied the kernel module files 'streams-*.o' to the directory '/lib/modules/2.4.20-28.7/strsock'. This means that to load any of these modules, you can simply execute, for example, 'modprobe stream-*somedriver*'.[37]

6.7.1.1 Linux Fast-STREAMS Module Loading

The 'strsock' demand load system supports both the old kerneld and the new kmod mechanisms for demand loading kernel modules.

The convention for 'strsock' kernel loadable object files is:

- Their name start with "streams-".
- They are placed in '/lib/modules/2.4.20-28.7/streams/', where '2.4.20-28.7' is an example kernel version.

If your kernel has been built using the 'kerneld' daemon, then 'strsock' kernel modules will automatically load as soon as the *STREAMS* module is pushed or the driver is opened. The 'make install' process makes the necessary changes to the '/etc/modules.conf' file. After the install, you will see lines like the following added to your '/etc/modules.conf' file:

```
prune modules.strsock
if -f /lib/modules/‘uname -r‘/modules.strsock
include /lib/modules/‘uname -r‘/modules.strsock
endif
```

which will provide for demand loading of the modules if they have been built and installed for the running kernel. The '/lib/modules/‘uname -r‘/modules.strsock' file looks like this:

```
alias char-major-245  streams-some_driver
alias char-major-246  streams-other_driver
```

Note that *STREAMS* modules are not listed in this file, but will be loaded by name using 'kerneld' if available.

Linux Fast-STREAMS has a wider range of kernel module loading mechanisms than is provided by the deprecated *LiS*. For mechanisms used for kernel module loading under *Linux Fast-STREAMS*, See Section "Top" in *Linux Fast-STREAMS Reference Manual*.

6.7.1.2 Linux STREAMS Module Loading

LiS is deprecated and this section has been deleted.

6.8 Maintenance

[37] Note that the '_kversion' of '2.4.20-28.7' is only an example.

6.8.1 Makefile Targets

automake(1) has many targets, not all of which are obvious to the casual user. In addition, *OpenSS7* automake(1) files have additional rules added to make maintaining and releasing a package somewhat easier. This list of targets provides some help with what targets can be invoked, what they do, and what they hope to achieve. The available targets are as follows:

6.8.1.1 User Targets

The following are normal targets intended to be invoked by installers of the package. They are concerned with compiling, checking the compile, installing, checking the installation, and removing the package.

'[all]' This is also the default target. It compiles the package and all release packages selected by 'configure'. This is performed after configuring the source with 'configure'. A 'Makefile' stub is provided so that if the package has not had autoreconf(1) run (such as when checked out from CVS, the package will attempt to run 'autoreconf -fiv'.

All *OpenSS7 Project* packages are configured without maintainer mode and without dependency tracking by default. This speeds compilation of the package for one-time builds. This also means that if you are developing using the source package (edit-compile-test cycle), changes made to source files will not cause the automatic rebuilding due to dependencies. There are two ways to enable dependency tracking: specify '--enable-maintainer-mode' to 'configure'; or, specify '--enable-dependency-tracking' to 'configure'. I use the former during my edit-compile-test cycle.

This is a standard *GNU* automake(1) makefile target. This target does not require root privilege.

'check' All *OpenSS7 Project* release packages provide check scripts for the check target. This step is performed after compiling the package and will run all of the 'check' programs against the compiled binaries. Which checks are performed depends on whether '--enable-maintainer-mode' was specified to 'configure'. If in maintainer mode, checks that assist with the release of the package will be run (such as checking that all manual pages load properly and that they have required sections.) We recommend running the check stage before installing, because it catches problems that might keep the installed package from functioning properly.

Another way to enable the greater set of checks, without invoking maintainer mode, is to specify '--enable-checks' to 'configure'. For more information, see Section 7.1.1 [Pre-installation Checks], page 89.

This is a standard *GNU* automake(1) makefile target, although the functions performed are customized for the *OpenSS7 Project*. This target does not require root privilege.

'install'
'install-strip'
The 'install' target installs the package by installing each release package. This target also performs some actions similar to the pre- and post-install scripts

used by packaging tools such as `rpm(1)` or `dpkg(1)`. The '`install-strip`' target strips unnecessary symbols from executables and kernel modules before installing.

This is a standard *GNU* `automake(1)` makefile target. This target requires root privilege.

'`installcheck`'
: All *OpenSS7 Project* packages provide test scripts for the '`installcheck`' target. Test scripts are created and run using `autotest` (part of the `autoconf(1)` package). Which test suites are run and how extensive they are depends on whether '`--enable-maintainer-mode`' was specified to '`configure`'. When in maintainer mode, all test suites will be run. When not in maintainer mode, only a few post-install checks will be performed, but the test suites themselves will be installed in '`/usr/libexec/strsock`'[38] for later use.

  This is a standard *GNU* `automake(1)` makefile target. This target might require root privilege. Tests requiring root privilege will be skipped when run as a regular user. Tests requiring regular account privileges will be skipped when run as root.

'`retest`'
: To complement the '`installcheck`' target above, all *OpenSS7 Project* packages provide the '`retest`' target as a means to rerun failed conformance test suite test cases. The '`retest`' target is provided because some test cases in the test suites have delicate timing considerations that allow them to fail sporadically. Invoking this target will retest the failed cases until no cases that are not expected failures remain.

  This is an *OpenSS7 Project* specific makefile target. As with '`installcheck`', this target might require root privilege. Tests requiring root privilege will be skipped when run as a regular user. Tests requiring regular account privileges will be skipped when run as root.

'`uninstall`'
: This target will reverse the steps taken to install the package. This target also performs pre- and post- erase scripts used by packaging tools such as *rpm* or *dpkg*. You need to have a configured build directory from which to execute this target, however, you do not need to have compiled any of the files in that build directory.[39]

  The '`uninstall`' target unfortunately removes add-on packages in the same order in which they were installed. This is not good for the *OpenSS7 Master Package*, where the '`remove`' target should be used instead.

  This is a standard *GNU* `automake(1)` makefile target. This target requires root privilege.

'`remove`'
: This target is like '`uninstall`' with the exception that it removes add-on packages in the reverse order that installation was performed.[40]

---

[38] '`/usr/libexec/strsock`' is just an example, the actual location is '`${libexecdir}/${PACKAGE}`', which varies from distribution to distribution (as some distributions such as Mandriva do not have a libexec directory).

[39] Therefore, it is possible to download the package, configure it, and then uninstall it. This is handy if you do not have the sources used to build and install the package immediately available.

[40] This is useful from the *OpenSS7 Master Package*.

This is an *OpenSS7 Project* specific makefile target. This target requires root privilege.

6.8.1.2 Maintainer Targets

The following targets are targets intended for use by maintainers of the package, or those responsible for release and packaging of a derivative work of the package. Some of these targets are only effective when maintainer mode has been invoked ('`--enable-maintainer-mode`' specified to '`configure`'.)

'`dist`' Creates a distribution package (tarball) in the top level build directory. *OpenSS7 Project* packages distribute two archives: a '`gzip tar`' archive and a '`bzip tar`' archive. These archives will have the name '`strsock-0.9.2.4.tar.gz`' and '`strsock-0.9.2.4.tar.bz2`'.

This is a standard *GNU* `automake(1)` makefile target. This target does not require root privilege.

'`distcheck`'
This target is intended for use when releasing the package. It creates the `tar(1)` archives above and then unpacks the tarball in a source directory, configures in a separate build directory, compiles the package, installs the package in a separate install directory, tests the install package to ensure that some components work, and, finally, uses the unpacked source tree to build another tarball. If you have added or removed files from the package, this is a good way to ensure that everything is still stable for release.

This is a standard *GNU* `automake(1)` makefile target. This target does not require root privilege.

6.8.1.3 Clean Targets

'`mostlyclean`'
Cleans out most of the files from the compile stage. This target is helpful if you have not enabled dependency tracking and need to recompile with changes.

This is a standard *GNU* `automake(1)` makefile target. This target does not require root privilege.

'`clean`' Cleans all the files from the build directory generated during the '`make [all]`' phase. It does not, however, remove files from the directory left there from the '`configure`' run. Use the '`distclean`' target to remove those too.

This is a standard *GNU* `automake(1)` makefile target. This target might require root privilege if the '`installcheck`' target or the `testsuite` was invoked with root privilege (leaving files belonging to root).

'`distclean`'
This target cleans out the directories left behind by '`distcheck`' and removes all the '`configure`' and generated files from the build directory. This will effectively remove all the files in the build directory, with the except of files that belong to you or some other process.

This is a standard *GNU* `automake(1)` makefile target. This target might require root privilege if the '`installcheck`' target or the `testsuite` was invoked with root privilege (leaving files belonging to root).

'`maintainer-clean`'
: This target not only removes files from the build directory, it removes generated files from the source directory as well. Care should be taken when invoking this target, because it removes files generated by the maintainer and distributed with the archive that might require special tools to regenerate. These special tools might only be available to the maintainer.[41] It also means that you probably need a full blown Linux system to rebuild the package. For more information, see Section 6.2.8 [Downloading from CVS], page 55.

  This is a standard *GNU* `automake(1)` makefile target. This target might require root privilege if the '`installcheck`' target or the `testsuite` was invoked with root privilege (leaving files belonging to root).

'`check-clean`'
: This target removes log files left behind by the '`check`' target. By default, the check scripts append to log files in the top level build directory. This target can be used to clean out those log files before the next run.

  This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

6.8.1.4 Manual Page Targets

The following targets are used to build, install and uninstall just the manual pages from the distribution. These targets are good for creating a distribution of just the manual pages. When building atop multiple packages, these targets recurse down through each package.

'`mans`'
: Build all of the manual pages. This involves performing parameter substitution on manual pages and optionally cooking the manual pages if '`--with-cooked-manpages`' was requested during configuration.

'`install-mans`'
: Installs the manual pages under *DESTDIR*. Specify *DESTDIR* to place the manual pages wherever you see fit. If *DESTDIR* is not specified on the command line, the manual pages will be installed in the normal installation directory.

'`uninstall-mans`'
: Uninstalls the manual pages from *DESTDIR*. Specify *DESTDIR* to indicate where to remove the manual pages from. If *DESTDIR* is not specified on the command line, the manual pages will be removed from the normal installation directory.

6.8.1.5 Release Targets

The following are targets used to generate complete releases into the package distribution directory. These are good for unattended and NFS builds, which is what I use them for.

[41] Theoretically this is true, however, the *OpenSS7 Project* does not use any maintainer programs that are not generally available (i.e. open source).

Also, when building from atop multiple packages, these targets also recurse down through each package.

'`release`' Build all of the things necessary to generate a release. On an `rpm(1)` system this is the distribution archives, the source rpm, and the architecture dependent and architecture independent binary rpms. All items are placed in the package distribution directory that can be specified with the '`--with-pkg-distdir=DIR`' option to '`configure`'.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

'`forced-release`'
The '`release`' target will not regenerate any files that already exist in the package distribution directory. This forced target will.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

'`release-sign`'
You will be prompted for a password, unless to specify it to make with the *GNUPGPASS* variable. For unattended or non-interactive builds with signing, you can do that as: '`make GNUPGPASS=mypasswd release-sign`'

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

'`forced-release-sign`'
The '`release-sign`' target will not regenerate any files that already exist in the package distribution directory. This forced target will.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

'`release-clean`'
This target will remove all distribution files for the current package from the package distribution directory.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

6.8.1.6 Logging Targets

For convenience, to log the output of a number of targets to a file, log targets are defined. The log file itself is used as the target to make, but make invokes the target minus a '`.log`' suffix. So, for example, to log the results of target '`foo`', invoke the target '`foo.log`'. The only target that this does not apply to is '`compile.log`'. When you invoke the target '`compile.log`' a simple `automake(1)` is invoked and logged to the file '`compile.log`'. The '`foo.log`' rule applies to all other targets. This does not work for all targets, just a selected few.[42] Following are the logging targets:

[42] Note that because logging targets invoke a pipe, `automake(1)` does not return the correct return status (always returns success if the `tee(1)` operation is successful). Therefore, these targets should not be invoked by scripts that need to use the return value from `automake(1)`.

Common Logging Targets

Common logging targets correspond to normal user `automake(1)` makefile targets as follows:

‘`compile.log`’
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake(1)` makefile target ‘`[all]`’.

‘`check.log`’
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake(1)` makefile target ‘`check`’.

‘`install.log`’
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake(1)` makefile target ‘`install`’.

‘`installcheck.log`’
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake(1)` makefile target ‘`installcheck`’.

‘`uninstall.log`’
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake(1)` makefile target ‘`uninstall`’.

‘`remove.log`’
: This is an *OpenSS7 Project* specific makefile target, that invokes the *OpenSS7 Project* ‘`remove`’ target.

Maintainer Logging Targets

Maintainer logging targets correspond to maintainer mode `automake(1)` makefile targets as follows:

‘`dist.log`’
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake(1)` makefile target ‘`dist`’.

‘`distcheck.log`’
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake(1)` makefile target ‘`distcheck`’.

‘`srpm.log`’
: This is an *OpenSS7 Project* specific makefile target, that invokes the *OpenSS7 Project* ‘`srpm`’ target.

‘`rebuild.log`’
: This is an *OpenSS7 Project* specific makefile target, that invokes the *OpenSS7 Project* ‘`rebuild`’ target.

‘`resign.log`’
: This is an *OpenSS7 Project* specific makefile target, that invokes the *OpenSS7 Project* ‘`resign`’ target.

‘`release.log`’
: This is an *OpenSS7 Project* specific makefile target, that invokes the *OpenSS7 Project* ‘`release`’ target.

‘`release-sign.log`’
: This is an *OpenSS7 Project* specific makefile target, that invokes the *OpenSS7 Project* ‘`release-sign`’ target.

If you want to add one, simply add it to *LOGGING_TARGETS* in ‘`Makefile.am`’.

6.8.1.7 Problem Report Targets

To ease problem report generation, all logging targets will automatically generate a problem report suitable for mailing in the file ‘*target*`.pr`’ for target ‘*target*`.log`’. This problem report file is in the form of an email and can be sent using the included `send-pr` script or by invoking the ‘`send-pr`’ makefile target.

There are two additional problem report targets:

‘`pr`’
: The ‘`pr`’ target is for independently generating a problem report outside of the build or installation process. The target will automatically generate a problem report skeleton suitable for editing and mailing in the file ‘`problem.pr`’. This problem report file is in the form of an email and can be edited and sent directly, or sent using the included `send-pr` script or by invoking the ‘`send-pr`’ target.

  This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

‘`send-pr`’
: The ‘`send-pr`’ target is for finalizing and mailing a problem report generated either inside or outside the build and installation process. The target will automatically finalize and mail the ‘`problem.pr`’ problem report if it has changed since the last time that ‘`send-pr`’ was invoked.

  This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege (unless the problem report file was generated as root).

6.8.1.8 Release Archive Targets

The following targets are used to generate and clean distribution archive and signature files. Whereas the ‘`dist`’ target affects archives in the top build directory, the ‘`release-archive`’ targets affects archives in the package distribution directory (either the top build directory or that specified with ‘`--with-pkg-distdir=DIR`’ to ‘`configure`’).

You can change the directory to which packages are distributed by using the ‘`--with-pkg-distdir=DIR`’ option to ‘`configure`’. The default directory is the top build directory.

‘`release-archives`’
: This target creates the distribution archive files if they have not already been created. This not only runs the ‘`dist`’ target, but also copies the files to the distribution directory, which, by default is the top build directory.

  The files generated are named:

  ‘`strsock-0.9.2.4.tar.gz`’ and ‘`strsock-0.9.2.4.tar.bz2`’

  You can change this distribution directory with the ‘`--with-pkg-distdir`’ option to ‘`configure`’. See ‘`./configure --help`’ for more details on options.

  This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

‘`release-sign-archives`’
: This target is like ‘`release-archives`’, except that it also signs the archives using a *GPG* detached signature. You will be prompted for a password unless you pass the *GNUPGPASS* variable to make. For automated or unattended builds, pass the *GNUPGPASS* variable like so:

  ‘`make GNUPGPASS=mypasswd release-sign-archives`’

  Signature files will be named:

  ‘`strsock-0.9.2.4.tar.gz.asc`’ and ‘`strsock-0.9.2.4.tar.bz2.asc`’

  These files will be moved to the package distribution directory with the plain text archives.

  This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

‘`release-clean-archives`’
: This target will clean the release archives and signature files from the package distribution directory.

  This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

6.8.1.9 RPM Build Targets

On `rpm(1)` systems, or systems sporting rpm packaging tools, the following targets are used to generate `rpm(1)` release packages. The epoch and release number can be controlled by the contents of the ‘`.rpmepoch`’ and ‘`.rpmrelease`’ files, or with the ‘`--with-rpm-epoch=EPOCH`’ and ‘`--with-rpm-release=RELEASE`’ options to ‘`configure`’. See ‘`configure --help`’ for more information on options. We always use release number ‘`1`’. You can use release numbers above ‘`1`’.

‘`srpm`’
: This target generates the source rpm for the package (without signing the source rpm). The source rpm will be named: ‘`strsock-0.9.2.4-1.srpm`’.

  This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

‘`rpms`’
: This target is responsible for generating all of the package binary rpms for the architecture. The binary rpms will be named:

  ‘`strsock-*-0.9.2.4-1.*.rpm`’

  where the stars indicate the subpackage and the architecture. Both the architecture specific subpackages (binary objects) and the architecture independent (‘`.noarch`’) subpackages will be built unless the the former was disabled with the option ‘`--disable-arch`’, or the later with the option ‘`--disable-indep`’, passed to ‘`configure`’.

  This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

‘`sign`’
‘`srpm-sign`’
: These two targets are the same. When invoked, they will add a signature to the source rpm file, provided that the file does not already have a signature.

You will be prompted for a password if a signature is required. Automated or unattended builds can be achieved by using the `emake` expect script, included in '`${srcdir}/scripts/emake`'.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

'`rebuild`' This target accepts searches out a list of kernel names from the '`${DESTDIR}/lib/modules`' directory and builds rpms for those kernels and for each of a set of architectures given in the *AM_RPMTARGETS* variable to make. This is convenience target for building a group of rpms on a given build machine.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

'`resign`' This target will search out and sign, with a *GPG* signature, the source rpm, and all of the binary rpms for this package that can be found in the package distribution directory. This target will prompt for a *GPG* password. Automated or unattended builds can be achieved with the `emake` expect script located here: '`${srcdir}/scripts/emake`'.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

6.8.1.10 Debian Build Targets

On Debian systems, or systems sporting Debian packaging tools, the following targets are used to generate Debian release packages. The release number can be controlled by the contents of the '`.debrelease`' file, or with the '`--with-debrelease=RELEASENUMBER`' option to '`configure`'. See '`configure --help`' for more information on options.

'`dsc`' This target will build the Debian source change package ('`.dsc`' file). We use release number '`0`' so that the entire tarball is included in the '`dsc`' file. You can use release number '`1`' for the same purposes. Release numbers above '`1`' will not include the entire tarball. The '`.dsc`' file will be named: '`strsock_0.9.2.4-0.dsc`'.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

'`sigs`' This target signs the '`.deb`' files. You will be prompted for a password, unless to specify it to make with the *GNUPGPASS* variable.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

'`debs`' This target will build the Debian binary package ('`.deb`' file) from the '`.dsc`' created above. (This target will also create the '`.dsc`' if it has not been created already.) The subpackage '`.deb`' files will be named: '`strsock-*_0.9.2.4-0_*.deb`', where the stars indicate the subpackage and the architecture.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

‘`csig`’ This target signs the ‘`.dsc`’ file. You will be prompted for a password, unless to specify it to make with the *GNUPGPASS* variable.

This is an *OpenSS7 Project* specific makefile target. This target does not require root privilege.

6.8.1.11 Documentation Targets

On systems that have `doxygen(1)` documentation tool, the following targets are used to generate doxygen html documentation:

‘`doxy`’ This target generates `doxygen(1)` documetnation from suitably marked sources. File containing the necessary documentation marks are discovered automatically by `configure`. Doxygen documentation can be generated bus is not distributed. Documentation is cerated in the subdirectory ‘`doc/html`’.

7 Troubleshooting

7.1 Test Suites

7.1.1 Pre-installation Checks

Most *OpenSS7* packages, including the *OpenSS7 STREAMS Sockets* package, ship with pre-installation checks integral to the build system. Pre-installation checks include check scripts that are shipped in the '`scripts`' subdirectory as well as specialized `make` targets that perform the checks.

When building and installing the package from *RPM* or *DEB* source packages (see Section 6.4.1 [Building from the Source RPM], page 73; and Section 6.4.2 [Building from the Debian DSC], page 74), a fundamental set of post-compile, pre-installation checks are performed prior to building binary packages. This is performed automatically and does not require any special actions on the part of the user creating binary packages from source packages.

When building and installing the package from *tarball* (see Section 6.4.3 [Building from the Tar Ball], page 74; and Section 6.5.3 [Installing the Tar Ball], page 76), however, pre-installation checks are only performed if specifically invoked by the builder of the package. Pre-installation checks are invoked after building the package and before installing the package. Pre-installation checks are performed by invoking the '`check`' or '`check.log`' target to `make` when building the package, as shown in Example 7.1.

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.bz2
% tar -xjvf strsock-0.9.2.4.tar.bz2
% pushd strsock-0.9.2.4
% ./configure
% make
% make check  # <------- invoke pre-installation checks
% popd
```

Example 7.1: *Invoking Pre-Installation Checks*

Pre-installation checks fall into two categories: *System Checks* and *Maintenance Checks*.

7.1.1.1 Pre-Installation System Checks

System Checks are post-compilation checks that can be performed before installing the package that check to ensure that the compiled objects function and will be successfully installed. When the '`--enable-maintainer-mode`' option has not been passed to `configure`, only *System Checks* will be performed.

For example, the steps shown in Example 7.2 will perform *System* checks.

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.bz2
% tar -xjvf strsock-0.9.2.4.tar.bz2
% pushd strsock-0.9.2.4
% ./configure
% make
% make check  # <------ invokes System pre-installation checks
% popd
```

Example 7.2: *Invoking System Checks*

7.1.1.2 Pre-Installation Maintenance Checks

Maintenance Checks include all *System Checks*, but also checks to ensure that the kernel modules, applications programs, header files, development tools, test programs, documentation, and manual pages conform to *OpenSS7* standards. When the '`--enable-maintainer-mode`' option has been passed to `configure`, *Maintenance Checks* will be performed.

For example, the steps shown in Example 7.3 will perform *Maintenance* checks.

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.bz2
% tar -xjvf strsock-0.9.2.4.tar.bz2
% pushd strsock-0.9.2.4
% ./configure --enable-maintainer-mode
% make
% make check  # <------ invokes Maintenance pre-installation checks
% popd
```

Example 7.3: *Invoking Maintenance Checks*

7.1.1.3 Specific Pre-Installation Checks

A number of check scripts are provided in the '`scripts`' subdirectory of the distribution that perform both *System* and *Maintenance* checks. These are as follows:

`check_commands`

This check performs both *System* and *Maintenance* checks.

When performing *System* tests, the following tests are performed:

Unless cross-compiling, or unless a program is included in `AM_INSTALLCHECK_STD_OPTIONS_EXEMPT` every program in `bin_PROGRAMS`, `sbin_PROGRAMS`, and `libexec_PROGRAMS` is tested to ensure that the '`--help`', '`--version`', and '`--copying`' options are accepted. When cross-compiling is is not possible to execute cross-compiled binaries, and these checks are skipped in that case.

Script executables, on the other hand, can be executed on the build host, so, unless listed in `AM_INSTALLCHECK_STD_OPTIONS_EXEMPT`, every program in `dist_bit_SCRIPTS`, `dist_sbin_SCRIPTS`, and `pkglibexec_SCRIPTS` are tested to ensure that the '`--help`', '`--version`', and '`--copying`' options are accepted.

When performing *Maintenance* tests, `check_commands` also checks to ensure that a manual page exists in section 1 for every executable binary or script

that will be installed from `bin_PROGRAMS` and `dist_bin_SCRIPTS`. It also checks to ensure that a manual page exists in section 8 for every executable binary or script that will be installed from `sbin_PROGRAMS`, `dist_sbin_SCRIPTS`, `libexec_PROGRAMS`, and `pkglibexec_SCRIPTS`.

`check_decls`

This check only performs *Maintenance* checks.

It collects the results from the `check_libs`, `check_modules` and `check_headers` check scripts and tests to ensure every declaration of a function prototype or external variable contained in installed header files has a corresponding exported symbol from either a to be installed shared object library or a to be installed kernel module. Declarations are exempted from this requirement if their identifiers have been explicitly added to the `EXPOSED_SYMBOL` variable. If `WARN_EXCESS` is set to '`yes`', then the check script will only warn when excess declarations exist (without a corresponding exported symbol); otherwise, the check script will generate an error and the check will fail.

`check_headers`

This check only performs *Maintenance* checks.

When performing *Maintenance* tests, it identifies all of the declarations included in to be installed header files. It then checks to ensure that a manual page exists in sections 2, 3, 7 or 9, as appropriate, for the type of declaration. It also checks to see if a manual page source file exists in the source directory for a declaration that has not been included in the distribution. Function or prototype declarations that do not have a manual page in sections 2, 3, or 9 will cause the check to fail. Other declarations ('`variable`', '`externvar`', '`macro`', '`enumerate`', '`enum`', '`struct`', '`union`', '`typedef`', '`member`', etc.) will only warn if a manual page does not exist, but will not fail the check.

`check_libs`

This check only performs *Maintenance* checks.

When performing *Maintenance* tests, it checks that each exported symbol in each to be installed shared object library has a manual page in section 3. It also checks that each exported symbol has a '`function`', '`prototype`' or '`externvar`' declaration in the to be installed header files. A missing declaration or manual page will cause this check to fail.

`check_mans`

This check only performs *Maintenance* checks.

When performing *Maintenance* tests, it checks that to be install manual pages can be formatted for display without any errors or warnings from the build host `man` program. It also checks that required headings exist for manual pages according to the section in which the manual page will be installed. It warns if recommended headings are not included in the manual pages. Because some *RPM* distributions have manual pages that might conflict with the package manual pages, this check script also checks for conflicts with installed manual pages on the build host. This check script also checks to ensure that all to be

installed manual pages are used in some fashion, that is, they have a declaration, or exported symbol, or are the name of a kernel module or STREAMS module or driver, possibly capitalized.

Note that checking for conflicts with the build host should probably be included in the *System* checks (because *System* checks are performed before the source *RPM* `%install` scriptlet).

`check_modules`

This check performs both *System* and *Maintenance* checks.

When performing *System* tests, it checks each to be installed kernel module to ensure that all undefined symbols can be resolved to either the kernel or another module. It also checks whether an exported or externally declared symbol conflicts with an exported or externally declared symbol present in the kernel or another module.[1]

When performing *Maintenance* tests, this check script tests that each to be installed kernel module has a manual page in section 9 and that each exported symbol that does not begin with an underscore, and that belongs to an exported function or exported variable, has a manual page in section 9. It also checks to ensure that each exported symbol that does not begin with an underscore, and that belongs to an exported function or exported variable, has a '`function`', '`prototype`' or '`externvar`' declaration in the to be installed header files.

`check_streams`

This check performs only *Maintenance* checks.

When performing *Maintenance* tests, it checks that for each configured *STREAMS* module or driver, or device node, that a manual page exists in section 4 or section 7 as appropriate.

The output of the pre-installation tests are fairly self explanatory. Each check script saves some output to '*`name`*`.log`', where *name* is the name of the check script as listed above. A summary of the results of the test are display to standard output and can also be captured to the '`check.log`' file if the '`check.log`' target is used instead of the '`check`' target to `make`.

Because the check scripts proliferate '*`name`*`.log`' files throughout the build directory, a '`make check-clean`' `make` target has be provided to clean them out. '`make check-clean`' should be run before each successive run of '`make check`'.

7.1.2 Post-installation Checks

Most OpenSS7 packages ship with a compatibility and conformance test suite built using the '`autotest`' capabilities of '`autoconf`'. These test suites act as a wrapper for the compatibility and conformance test programs that are shipped with the package.

Unlike the pre-installation checks, the post-installation checks are always run complete. The only check that post-installation test scripts perform is to test whether they have been invoked with root privileges or not. When invoked as root, or as a plain user, some tests might be skipped that require root privileges, or that require plain user privileges, to complete successfully.

[1] This particular check has caught some name space pollution that has occurred in the 2.6.11 kernel.

7.1.2.1 Running Test Suites

There are several ways of invoking the conformance test suites:

1. The test suites can be run after installation of the package by invoking the '`make installcheck`' or '`make installcheck.log`' target. Some packages require that root privileges be acquired before invoking the package.
2. The test suites can be run from the distribution subdirectory after installation of the package by invoking the `testsuite` shell script directly.
3. The test suites can be run standalone from the '`libexec`' ('`/usr/libexec`') installation directory by invoking the `testsuite` shell script directly.

Typical steps for invoking the test suites directly from `make` are shown in Example 7.4.

```
% wget http://www.openss7.org/strsock-0.9.2.4.tar.bz2
% tar -xjvf strsock-0.9.2.4.tar.bz2
% pushd strsock-0.9.2.4
% ./configure
% make
% make check  # <------ invokes System pre-installation checks
% make install
% sudo make installcheck # <------- invokes post-installation tests
% popd
```

Example 7.4: *Invoking System Checks*

When performing post-installation checks for the purposes of generating a problem report, the checks should always be performed from the build directory, either with '`make installcheck`' or by invoking `testsuite` directly from the '`tests`' subdirectory of the build directory. This ensures that all of the information known to `configure` and pertinent to the configuration of the system for which a test case failed, will be collected in the resulting '`testsuite.log`' file deposited upon test suite failure in the '`tests`' directory. This '`testsuite.log`' file can then be attached as part of the problem report and provides rich details to maintainers of the package. See also See Section 7.2 [Problem Reports], page 93, below.

Typical steps for invoking and installed `testsuite` standalone are shown in Example 7.5.

```
% [sudo] /usr/libexec/strsock/testsuite
```

Example 7.5: *Invoking* `testsuite` *Directly*

When invoked directly, `testsuite` will generate a '`testsuite.log`' file in the current directory, and a '`testsuite.dir`' directory of failed tests cases and debugging scripts. For generating a problem report for failed test cases, see Section 7.2.4 [Stand Alone Problem Reports], page 96.

7.2 Problem Reports

7.2.1 Problem Report Guidelines

Problem reports in the following categories should include a log file as indicated in the table below:

‘`./configure`’
: A problem with the configuration process occurs that causes the ‘`./configure`’ command to fail. The problem report must include the ‘`config.log`’ file that was generated by `configure`.

‘`make compile.log`’
: A problem with the build process occurs that causes the ‘`make`’ command to fail. Perform ‘`make clean`’ and then ‘`make compile.log`’ and attach the ‘`config.log`’ and ‘`compile.log`’ files to the problem report.

‘`make check.log`’
: A problem occurs with the ‘`make check`’ target that causes it to fail. Perform ‘`make check-clean check.log`’ and attach the ‘`config.log`’, ‘`compile.log`’ and ‘`check.log`’ files to the problem report.

‘`sudo make install.log`’
: A problem occurs with ‘`sudo make install`’ that causes it to fail. Perform ‘`sudo make uninstall`’ and ‘`sudo make install.log`’ and attach the ‘`config.log`’, ‘`compile.log`’, ‘`check.log`’, and ‘`install.log`’ files to the problem report.

‘`[sudo] make installcheck.log`’
: A problem occurs with the ‘`make installcheck`’ target that causes the test suite to fail. Attach the resulting ‘`tests/testsuite.log`’ and ‘`installcheck.log`’ file to the problem report. There is no need to attach the other files as they are included in ‘`tests/testsuite.log`’.

‘`[sudo] make uninstall.log`’
: A problem occurs with the ‘`make uninstall`’ target that causes the test suite to fail. Perform ‘`sudo make uninstall.log`’ and attach the ‘`config.log`’, ‘`compile.log`’, ‘`check.log`’, ‘`install.log`’, ‘`installcheck.log`’, ‘`tests/testsuite.log`’ and ‘`uninstall.log`’ file to the problem report.

‘`[sudo] make remove.log`’
: A problem occurs with the ‘`make remove`’ target that causes the test suite to fail. Perform ‘`sudo make remove.log`’ and attach the ‘`config.log`’, ‘`compile.log`’, ‘`check.log`’, ‘`install.log`’, ‘`installcheck.log`’, ‘`tests/testsuite.log`’ and ‘`remove.log`’ file to the problem report.

For other problems that occur during the use of the *OpenSS7 STREAMS Sockets* package, please write a test case for the test suite that recreates the problem if one does not yet exist and provide a test program patch with the problem report. Also include whatever log files are generated by the kernel (`cmn_err(9)`) or by the `strerr(8)` or `strace(1)` facilities (`strlog(9)`).

7.2.2 Generating Problem Reports

The OpenSS7 Project uses the *GNU GNATS* system for problem reporting. Although the ‘`send-pr`’ tool from the *GNU GNATS* package can be used for bug reporting to the project’s

GNATS database using electronic mail, it is not always convenient to download and install the *GNATS* system to gain access to the '`send-pr`' tool.

Therefore, the *OpenSS7 STREAMS Sockets* package provides the '`send-pr`' shell script that can be used for problem reporting. The '`send-pr`' shell script can invoked directly and is a work-alike for the *GNU* '`send-pr`' tool.

The '`send-pr`' tool takes the same flags and can be used in the same fashion, however, whereas '`send-pr`' is an interactive tool[2], '`send-pr`' is also able to perform batch processing. Whereas '`send-pr`' takes its field information from local databases or from using the '`query-pr`' C-language program to query a remote database, the '`send-pr`' tool has the field database internal to the tool.

Problem reports can be generate using `make`, See Section 6.8.1.7 [Problem Report Targets], page 85. An example of how simple it is to generate a problem report is illustrated in Example 7.6.

```
% make pr
SEND-PR:
SEND-PR: send-pr:  send-pr was invoked to generate an external report.  An
SEND-PR: automated problem report has been created in the file named
SEND-PR: 'problem.pr' in the current directory.  This problem report can
SEND-PR: be sent to bugs@openss7.org by calling this script as
SEND-PR: '/home/brian/os7/scripts/send-pr --file="problem.pr"'.
SEND-PR:
SEND-PR: It is possible to edit some of the fields before sending on the
SEND-PR: problem report.  Please remember that there is NO WARRANTY.  See
SEND-PR: the file 'COPYING' in the top level directory.
SEND-PR:
SEND-PR: Please do not send confidential information to the bug report
SEND-PR: address.  Inspect the file 'problem.pr' for confidential
SEND-PR: information before mailing.
SEND-PR:
% vim problem.pr  # <--- follow instructions at head of file
% make send-pr
```

Example 7.6: *Invoking Problem Report Generation*

Using the '`make pr`' target to generate a problem report has the advantages that it will assemble any available '`*.log`' files in the build directory and attach them to the problem report.

7.2.3 Automatic Problem Reports

The *OpenSS7 STREAMS Sockets* package also provides a feature for automatic problem report generation that meets the problem report submission guidelines detailed in the preceding sections.

Whenever a logging makefile target (see Section 6.8.1.6 [Logging Targets], page 83) is invoked, if the primary target fails, the `send-pr` shell script is invoked to automatically

[2] '`send-pr`' launches the user's *EDITOR* to edit the problem report before submitting it.

generate a problem report file suitable for the corresponding target (as described above under see Section 7.2.1 [Problem Report Guidelines], page 94). An example is shown in Example 7.7.

```
% make compile.log
...
...
make[5]: *** [libXNSdrvs_a-ip.o] Error 1
make[5]: Leaving directory `/u6/buildel4/strxns'
make[4]: *** [all-recursive] Error 1
make[4]: Leaving directory `/u6/buildel4/strxns'
make[3]: *** [all] Error 2
make[3]: Leaving directory `/u6/buildel4/strxns'
make[2]: *** [all-recursive] Error 1
make[2]: Leaving directory `/u6/buildel4'
make[1]: *** [all] Error 2
make[1]: Leaving directory `/u6/buildel4'
SEND-PR:
SEND-PR: send-pr:  Make target compile.log failed in the compile stage.  An
SEND-PR: automated problem report has been created in the file named
SEND-PR: 'problem.pr' in the current directory.  This problem report can
SEND-PR: be sent to bugs@openss7.org by calling 'make send-pr'.
SEND-PR:
SEND-PR: It is possible to edit some of the fields before sending on the
SEND-PR: problem report.  Please remember that there is NO WARRANTY.  See
SEND-PR: the file 'COPYING' in the top level directory.
SEND-PR:
SEND-PR: Please do not send confidential information to the bug report
SEND-PR: address.  Inspect the file 'problem.pr' for confidential
SEND-PR: information before mailing.
SEND-PR:
% vim problem.pr  # <--- follow instructions at head of file
% make send-pr
```

Example 7.7: *Problem Report from Failed Logging Target*

7.2.4 Stand Alone Problem Reports

The *OpenSS7 STREAMS Sockets* package installs the `send-pr` script and its configuration file '`send-pr.config`' in '`${libexecdir}/strsock`' along with the validation `testsuite`, see See Section 7.1 [Test Suites], page 89. As with the `testsuite`, this allows the `send-pr` script to be used for problem report generation on an installed system that does not have a build directory.

An example of invoking the package `testsuite` and then generating a problem report for failed cases is shown in Example 7.8.

```
% [sudo] /usr/libexec/strsock/testsuite
% # test cases failed...
% /usr/libexec/strsock/send-pr
SEND-PR:
SEND-PR: send-pr:  send-pr was invoked to generate an external report.  An
SEND-PR: automated problem report has been created in the file named
SEND-PR: 'problem.pr' in the current directory.  This problem report can
SEND-PR: be sent to bugs@openss7.org by calling this script as
SEND-PR: '/usr/libexec/strsock/send-pr --file problem.pr'.
SEND-PR:
SEND-PR: It is possible to edit some of the fields before sending on the
SEND-PR: problem report.  Please remember that there is NO WARRANTY.  See
SEND-PR: the file 'COPYING' in the top level directory.
SEND-PR:
SEND-PR: Please do not send confidential information to the bug report
SEND-PR: address.  Inspect the file 'problem.pr' for confidential
SEND-PR: information before mailing.
SEND-PR:
% vim problem.pr  # <--- follow instructions at head of file
% /usr/libexec/strsock/send-pr --file problem.pr
```

Example 7.8: *Invoking* `send-pr` *Directly*

The advantage of the approach shown in the example is that the `send-pr` script is capable of collecting the '`testsuite.log`' file and the failed test cases and debugging scripts from the '`testsuite.dir`' directory and including them in the problem report, as well as all package pertinent information from the installed '`send-pr.config`'.

7.3 Known Problems

The OpenSS7 Project does not ship software with known bugs. All bugs are unknown.

Verified behaviour is that behaviour that has been verified by conformance test suites that are shipped with the *OpenSS7 STREAMS Sockets* package.

Unverified behaviour may contain unknown bugs.

Please remember that there is **NO WARRANTY**.

See also Section 5.5 [Bugs], page 40, or file '`BUGS`' in the release directory.

Appendix A Possible BSD Sources

It is possible that some of the software contained in the following files might be ultimately derived from the associated *BSD Net2* sources:

OpenSS7 File	*BSD Net2 File*
‘src/drivers/socksys.c’	‘usr/src/uts/i386/io/osocket.c’
‘src/modules/sockmod.c’	‘usr/src/uts/i386/io/sockmod.c’
‘src/include/sys/sockmod.h’	‘usr/src/uts/i386/sys/sockmod.h’
‘src/include/sys/protosw.h’	‘usr/src/uts/i386/sys/protosw.h’
‘src/include/sys/socksys.h’	‘usr/src/uts/i386/sys/osocket.h’
‘src/include/sys/socket.h’	‘usr/src/uts/i386/sys/socket.h’
‘src/include/sys/socketvar.h’	‘usr/src/uts/i386/sys/socketvar.h’
‘src/include/sys/sockio.h’	‘usr/src/uts/i386/sys/sockio.h’
‘src/lib/sockets.c’	‘usr/src/lib/libsocket/socket/_conn_util.c’
‘src/lib/sockets.c’	‘usr/src/lib/libsocket/socket/_utility.c’
‘src/lib/accept.c’	‘usr/src/lib/libsocket/socket/accept.c’
‘src/lib/bind.c’	‘usr/src/lib/libsocket/socket/bind.c’
‘src/lib/connect.c’	‘usr/src/lib/libsocket/socket/connect.c’
‘src/lib/getpeername.c’	‘usr/src/lib/libsocket/socket/getpeernm.c’
‘src/lib/getsockname.c’	‘usr/src/lib/libsocket/socket/getsocknm.c’
‘src/lib/getsockopt.c’	‘usr/src/lib/libsocket/socket/getsockopt.c’
‘src/lib/listen.c’	‘usr/src/lib/libsocket/socket/listen.c’
‘src/lib/receive.c’	‘usr/src/lib/libsocket/socket/receive.c’
‘src/lib/ioctl.c’	‘usr/src/lib/libsocket/socket/s_ioctl.c’
‘src/lib/send.c’	‘usr/src/lib/libsocket/socket/send.c’
‘src/lib/setpeername.c’	‘usr/src/lib/libsocket/socket/setpeernm.c’
‘src/lib/setsockname.c’	‘usr/src/lib/libsocket/socket/setsocknm.c’
‘src/lib/setsockopt.c’	‘usr/src/lib/libsocket/socket/setsockopt.c’
‘src/lib/shutdown.c’	‘usr/src/lib/libsocket/socket/shutdown.c’
‘src/lib/socket.c’	‘usr/src/lib/libsocket/socket/socket.c’
‘src/lib/socketpair.c’	‘usr/src/lib/libsocket/socket/socketpair.c’

The files on the right were listed in *Exhibit C* of *The 1994 USL-Regents of UCal Settlement Agreement* and are therefore possibly subject to the *USL/BSD Combined License* of *Exhibit F* (see [BSD/USL Combined License], page 113).

Licenses

GNU Affero General Public License

The GNU Affero General Public License.
Version 3, 19 November 2007

Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are 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.

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.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

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 Affero 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. Remote Network Interaction; Use with the GNU General Public License.

    Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

    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 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 work with which it is combined will remain governed by version 3 of the GNU General Public License.

14. Revised Versions of this License.

    The Free Software Foundation may publish revised and/or new versions of the GNU Affero 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 Affero 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 Affero 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 Affero 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 Affero 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
Affero General Public License for more details.

You should have received a copy of the GNU Affero 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 your software can interact with users remotely through a network, you should also make sure that it provides a way for users to get its source. For example, if your program is a web application, its interface could display a "Source" link that leads users to an archive of the code. There are many ways you could offer source, and different solutions will be better for different programs; see section 13 for the specific requirements.

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 AGPL, see http://www.gnu.org/licenses/.

BSD/USL Combined License

BSD/USL COMBINED LICENSE

All or some portions of this file are derived from material licensed to the *University of California* by *American Telephone and Telegraph Co.* or *UNIX System Laboratories, Inc.* and are reproduced herein with the permission of *UNIX System Laboratories, Inc.*

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 *University of California, Berkeley*, and its contributors.

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.

GNU General Public 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.

GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Terms and Conditions

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

   As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

   "The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

   An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

   A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

   The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

   The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

   You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

   If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

   a. under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b. under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

   The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

   a. Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
   b. Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

   You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

   a. Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
   b. Accompany the Combined Work with a copy of the GNU GPL and this license document.
   c. For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
   d. Do one of the following:
      0. Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
      1. Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
   e. Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

   You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

   a. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

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

6. Revised Versions of the GNU Lesser General Public License.

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

   Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

   If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

END OF TERMS AND CONDITIONS

GNU Free Documentation License

GNU FREE DOCUMENTATION LICENSE
Version 1.1, March 2000

Preamble

The purpose of this License is to make a manual, textbook, or other written document *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.

Terms and Conditions for Copying, Distribution and Modification

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

   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. (For example, 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.

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 "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, 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 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.

2. 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.

3. COPYING IN QUANTITY

   If you publish printed copies 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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.

4. 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:

   A. 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.
   B. 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 less than five).
   C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
   D. Preserve all the copyright notices of the Document.
   E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
   F. 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.
   G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
   H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and 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.

J. 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.

K. In any section entitled "Acknowledgments" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.

L. 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.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

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.

5. 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.

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 "Acknowledgments", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. 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.

7. 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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. 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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

10. 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 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.

END OF TERMS AND CONDITIONS

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:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being list.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being *list*"; likewise for Back-Cover Texts.

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.

Indices

Index of Concepts

M

O

P

Q

R

S

T

U

W

Y

Index of Data Types

A

Y

Index of Functions and Macros

Index of Variables and Constants

Index of Files and Programs

/

S

Index of Configuration Options

Index of Makefile Targets

Index of Authors

Index of Manual Pages Referenced

S

T

U

W

X

Y

Z