OpenSS7 Installation and Reference Manual

OpenSS7
Installation and Reference Manual

Version 1.1 Edition 7.20141001
Updated October 25, 2014
Distributed with Package openss7-1.1.7.20141001

Abstract:

This document is a Installation and Reference Manual containing technical details concerning the implementation of the OpenSS7 for OpenSS7. It contains recommendations on software architecture as well as platform and system applicability of the OpenSS7.

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

Notice:

Short Contents

Table of Contents

Preface

Notice

This package is released and distributed under the *AGPL* (see [GNU Affero General Public License], page 201). 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.

This manual is released under the *FDL* (see [GNU Free Documentation License], page 211) with no invariant sections, no front-cover texts and no back-cover texts.

Abstract

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

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*.

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* 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* 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.

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.

Revision History

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.

[1] Formerly X/Open and UNIX International.

A current version of this manual is normally distributed with the *OpenSS7* package, `openss7-1.1.7.20141001`.[2]

Version Control

Although the author has attempted to ensure that the information in this document is complete and correct, neither the Author nor OpenSS7 Corporation will take any responsibility in it. *OpenSS7 Corporation* is making this documentation available as a reference point for the industry. While *OpenSS7 Corporation* believes that these interfaces are well defined in this release of the document, minor changes may be made prior to products conforming to the interfaces being made available. *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.

```
$Log: STREAMS.texi,v $
Revision 1.1.2.3  2011-02-07 02:21:33  brian
- updated manuals

Revision 1.1.2.2  2010-11-28 13:41:16  brian
- documentation updates

Revision 1.1.2.1  2009-06-21 10:40:08  brian
- added files to new distro
```

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 or contract, negligence or other tortious action, arising out of or in connection with any use of this documentation or the performance or implementation of the contents thereof.**

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

[2] http://www.openss7.org/repos/tarballs/openss7-1.1.7.20141001.tar.bz2

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

The OpenSS7 Project was funded in part by:

- Monavacon Limited
- OpenSS7 Corporation

Thanks to the subscribers to and sponsors of The OpenSS7 Project. Without their support, open software like this would not be possible.

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*, the *Linux Kernel Community*, and the open source software movement at large.

Sponsors

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

- Monavacon Limited
- OpenSS7 Corporation

Additional funding for The OpenSS7 Project was provided by:

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

Contributors

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

- Per Berquist
- John Boyd
- Chuck Winters
- Peter Courtney
- Tom Chandler
- Gurol Ackman
- Pierre Crepieux
- Christopher Lydick
- Omer Tunali
- John Hodgkinson
- Kutluk Testicioglu
- John Wenker
- Angel Diaz
- Jérémy Compostella
- Sylvain Chouleur
- Christophe Nolibos
- Bryan Shupe
- D. Milanovic
- Tony Abo
- Others

Supporters

Over the years a number of organizations have provided continued support in the form of assessment, inspection, testing, validation and certification.

Telecommunications

- Integrated Telecom Solutions
- Accuris Networks
- Adax
- AirNet Communications
- Alacre
- Alcatel-Lucent
- Anam
- Alaska Power & Telephone
- Artesyn (now Emerson)
- Bharti Telesoft
- Continuous Computing (Trillium)
- Cisco
- Cogeco Cable Inc.
- Condor Networks
- Corecess
- Cosini
- Datacraft
- Datatronics
- Digium
- DTAG (Deutsche Telecom AG)
- Engage Communication Inc.
- eServGlobal (NZ) Pty Ltd.
- Excel Telecommunications
- France Telecom
- Geolink (now SeaMobile)
- Huawei
- Integral Access (now Telco Systems)
- Kineto Wireless
- Maestro Communications
- Mindspeed
- Mobixell
- Motorola
- m-Wise Inc.
- Net2Phone
- NetTest A/S (now Anritsu)
- Newnet Communications
- Noble Systems Corporation
- Nortel Networks
- OnMobile
- Ouroboros
- Primal Technologies Inc.
- Peformance Technologies
- Reliance Communications
- SONORYS Technology GmbH
- AASTRA
- Aculab
- AEPONA
- Airwide Solutions
- Alcatel
- Altobridge
- Apertio (now Nokia)
- Aricent
- Arthus Technologies
- BubbleMotion
- Cellnext Solutions Limited
- Codent Networks
- Comverse Ltd.
- Coral Telecom
- Corelatus
- Data Connection
- Datatek Applications Inc.
- Dialogic
- Druid Software
- Empirix
- Ericsson
- ETSI
- Flextronics (now Aricent)
- Gemini Mobile Technologies
- Global Edge
- IBSYS Canada
- Integrat Mobile Aggregation Services
- Lucent
- MCI
- Mobis
- Motivity Telecom
- Mpathix Inc.
- Myriad Group
- NetCentrex S. A.
- NeuvaTel PCS
- NMS (now Dialogic)
- Nokia
- j2 Global Communications
- Orange
- P3 Solutions GmbH
- Propolys Pte Ltd.
- Pulse Voice Inc.
- Roamware Inc.
- Sonus Networks Inc.

- Spider Ltd. (now Emerson)
- Oasis Systems
- Stratus Technologies Bermuda Ltd.
- Switchlab Ltd.
- SysMaster Corporation
- Tecore
- Telcordia
- Teledesign
- Telnor
- Texas Instruments Inc.
- Ulticom
- Vecto Communications SRL
- VeriSign
- VSE NET GmbH
- WINGcon GmbH
- Xentel Inc.
- ZTE Corporation
- SS8 Networks Inc.
- Stratus
- Sicap AG
- Synapse Mobile Networks SA
- Tata Communications
- Tekno Telecom LLC
- Telecom Italia
- Telemetrics Inc.
- TE-Systems
- Tumsan Oy
- Vanu Inc.
- Veraz Networks
- Vodare Ltd.
- The Software Group Limited
- Wipro Technologies
- YCOM SA

Aerospace and Military

- Advanced Technologies
- Altobridge
- ARINC
- ATOS Origin
- Boeing
- Boldon James
- CRNA
- DSNA-DGAC[3]
- DLR[4]
- DSNA-DTI
- Egis-Avia (Sofreavia)
- MetaSlash
- Sofreavia
- FAA WJHTC[5]
- Thales ATM/Air Systems
- Altobridge
- BBN (Bolt, Beranek, and Neuman)
- Boldon James
- Lockheed Martin Co.
- Northrop Grumman Corporation
- QinetiQ
- SAAB
- Sandia National Laboratories
- Thales
- Wright-Patterson Air Force Base

Financial, Business and Security

- Alebra
- Automated Trading Desk (now Citi)
- Banco Credicoop
- BeMac
- Boldon James
- CyberSource Corporation
- Fujitsu-Seimens
- FutureSoft
- Gcom
- Alebra
- Boldon James
- Fujitsu-Seimens
- FutureSoft
- GSX
- HOB International
- HP (Hewlett-Packard)
- IBM

[3] La Direction des Services de la Navigation Aérienne - La Direction Général de l'Aviation Civile
[4] Deutsches Zentrum für Luft- unde Raumfarht
[5] Federal Aviation Administration - William J. Hughes Technical Center

- GSX
- HOB International
- HP (Hewlett-Packard)
- IBM
- Lightbride (now CyberSource)
- MasterCard
- Network Executive Software Inc.
- Packetware Inc.
- Packetware Inc.
- Prism Holdings Ltd.
- S2 Systems (now ACI)
- Symicron Computer Communications Limited
- Alert Logic
- Apani
- BeMac
- ERCOM
- Hitech Systems
- iMETRIK
- Intrado Inc.

Education, Health Care and Nuclear Power

- IEEE Computer Society
- ENST[6]
- HTW-Saarland[7]
- Kansas State University
- University of North Carolina Charlotte
- Ateb
- Mandexin Systems Corporation
- Areva NP
- European Organization for Nuclear Research

Agencies

It would be difficult for the OpenSS7 Project to attain the conformance and certifications that it has without the free availability of specifications documents and standards from standards bodies and industry associations. In particular, the following:

- 3GPP (Third Generation Partnership Project)
- ATM Forum
- EIA/TIA (Electronic Industries Alliance)
- ETSI (European Telecommunications Standards Institute)
- ICAO (International Civil Aviation Organization)
- IEEE (Institute of Electrical and Electronic Engineers)
- IETF (The Internet Engineering Task Force)
- ISO (International Organization for Standardization)
- ITU (International Telecommunications Union)
- Mulutiservices Forum
- The Open Group

Of these, ICAO, ISO, IEEE and EIA have made at least some documents publicly available. ANSI is notably missing from the list: at one time draft documents were available from ANSI (ATIS), but that was curtailed some years ago. Telecordia does not release any standards publicly. Hopefully these organizations will see the light and realize, as the others have, that to remain current as a standards organization in today's digital economy requires providing individuals with free access to documents.

Authors

The authors of the *OpenSS7* package include:

[6] Ecole Nationale Supérieure des Télécommunications

[7] Hochschule für Technik und Wirtschaft des Saarlandes

– Brian Bidulock

See ⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩, for a complete listing and cross-index of authors to sections of this manual.

Maintainer

The maintainer of the *OpenSS7* 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 9.2 [Problem Reports], page 195.

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* package, only after reading the `BUGS` file in the release, or See Section 9.2 [Problem Reports], page 195. 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* 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

Package openss7-1.1.7.20141001 was released under AGPLv3 2014-10-25.

This is the *OpenSS7* package. It contains all of the OpenSS7 Project release code. This is the only package released by the OpenSS7 Project.

The package contains all of the former subpackages of the OpenSS7 Master Package, including:

- Linux Fast-STREAMS
- STREAMS Compatibility Modules
- STREAMS Utilities
- STREAMS Terminals
- STREAMS X/Open Networking Services
- STREAMS X/Open Networking XTI/TLI Library
- STREAMS Sockets
- STREAMS Internet Protocol Drivers
- STREAMS SCTP
- STREAMS Channels and Multiplexing
- STREAMS X.25 and Frame Relay Stack
- STREAMS Open Systems Interconnect (OSI) Stack
- STREAMS ISDN Stack
- STREAMS ATM Stack
- STREAMS SS7 Stack
- STREAMS SIGTRAN Stack
- STREAMS VoiP Stack

Fully deprecated by this release and no longer released by the OpenSS7 Project are the following former subpackages of the OpenSS7 Master Package:[1]

- Linux Native Sockets SCTP
- Internet Performance (the OpenSS7 iperf fork)
- Network Performance (the OpenSS7 netperf fork)
- Dialogic Open System Release 6.1

This distribution is applicable to *Linux* 2.4, 2.6 and 3.x kernels as was targeted at `ix86`, `x86_64`, `ppc` and `ppc64` architectures, but should build and install for other architectures as well.

Release

This is the `openss7-1.1.7.20141001` package, released 2014-10-25. This '`1.1.7.20141001`' 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/openss7-1.1.7.20141001.tar.xz
```

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)`, `zypper(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 `openss7` package page for tarballs, source and binary packages.

[1] With the exception of Linux Native Sockets SCTP, these packages can be obtained from the projects or companites that originated them.

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-openss7 and README-make, files (or see Chapter 8 [Installation], page 109) for installation instructions.

When working from cvs(1) or git(1), please see the README-cvs, file (or see Section 8.2.10 [Downloading from CVS], page 128). An abbreviated installation procedure that works for most applications appears below.

This release of the package is published strictly under Version 3 of the *Affero GNU Public License* which can be found in the file COPYING. Package specific licensing terms (if any) can be found in the file LICENSES. Please respect these licensing arrangements. If you are interested in different licensing terms, please contact the copyright holder, or OpenSS7 Corporation <sales@openss7.com>.

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

Repository Installation

The simplest way of installing the package is to use the *OpenSS7* `repomd` or `apt` repositories instead of attempting to build from tarball. When you already have a the repository set up, the package can be udpated simply with (one of):

```
Fedora:   $> sudo yum update openss7
CentOS:   $> sudo yum update openss7
RHEL:     $> sudo yum update openss7
OpenSUSE: $> sudo zypper update openss7
SLES:     $> sudo zypper update openss7
Mageia:   $> sudo uprmi openss7
Mandriva: $> sudo urpmi openss7
MES:      $> sudo urpmi openss7
Ubuntu:   $> sudo aptitude install openss7
Debian:   $> sudo aptitude install openss7
```

If you have not yet set up an installation source for the *OpenSS7* repositories, one of the following commands will establish repository access for RPM-based systems:

```
          $> REPO=http://www.openss7.org/repo/rpms
Fedora:   $> SUBDIR=fedora/15/$(uname -m)/RPMS/noarch
CentOS:   $> SUBDIR=centos/5.6/$(uname -m)/RPMS/noarch
RHEL:     $> SUBDIR=redhat/6.0/$(uname -m)/RPMS/noarch
OpenSUSE: $> SUBDIR=opensuse/11.4/$(uname -m)/RPMS/noarch
SUSE:     $> SUBDIR=suse/11.1/$(uname -m)/RPMS/noarch
Mageia:   $> SUBDIR=mageia/1/$(uname -m)/RPMS/noarch
Mandriva: $> SUBDIR=mandriva/2011.0/$(uname -m)/RPMS-noarch
MES:      $> SUBDIR=mes/5.2/$(uname -m)/RPMS/noarch
          $> sudo rpm -Uhv $REPO/$SUBDIR/openss7-repo.noarch.rpm
```

For DPKG-based systems, one of the following command sequences will establish repository access for DPKG-based systems:

```
          $> REPO=http://www.openss7.org/repo/debs/
Debian:   $> SUBDIR=debian/squeeze/$(dpkg-architecture -a -qDEB_HOST_ARCH)/main
Ubuntu:   $> SUBDIR=ubuntu/10.04/$(dpkg-architecture -a -qDEB_HOST_ARCH)/main
          $> wget $REPO/$SUBDIR/openss7-repo_all.deb
          $> dpkg -i -D010077 openss7-repo_all.deb
```

For zypper(8) based systems it may be first necessary to perform:

```
OpenSUSE: $> sudo rpm --import https://www.openss7.org/pubkey.asc
SLES:     $> sudo rpm --import https://www.openss7.org/pubkey.asc
```

For apt(8) based systems it may be first necessary to perform:

```
          $> wget https://www.openss7.org/pubkey.asc
Debian:   $> sudo apt-key add pubkey.asc
Ubuntu:   $> sudo apt-key add pubkey.asc
```

For yum(8) based systems, caches should be refreshed with:

```
Fedora:   $> sudo yum makecache
CentOS:   $> sudo yum makecache
RHEL:     $> sudo yum makecache
```

For zypper(8) based systems, caches should be refreshed with:

```
OpenSUSE: $> sudo zypper refresh-services
OpenSUSE: $> sudo zypper refresh
SLES:     $> sudo zypper refresh-services
SLES:     $> sudo zypper refresh
```

For urpmi(8) based systems, caches should be refreshed with:

```
Mageia:   $> sudo urpmi.update
Mandriva: $> sudo urpmi.update
MES:      $> sudo urpmi.update
```

For apt(8) base systems, caches should be refreshed with:[2]

```
Debian:   $> sudo aptitude update
Ubuntu:   $> sudo aptitude update
```

Once the repository is set up in this fashion, it should be possible to install using (one of):

```
Fedora:   $> sudo yum install openss7
CentOS:   $> sudo yum install openss7
RHEL:     $> sudo yum install openss7
OpenSUSE: $> sudo zypper install openss7
SLES:     $> sudo zypper install openss7
Mageia:   $> sudo uprmi openss7
Mandriva: $> sudo urpmi openss7
MES:      $> sudo urpmi openss7
Debian:   $> aptitude install openss7
Ubuntu:   $> aptitude install openss7
```

The entire process can be undone with:

```
Fedora:   $> sudo yum remove openss7 openss7-repo
CentOS:   $> sudo yum remove openss7 openss7-repo
RHEL:     $> sudo yum remove openss7 openss7-repo
OpenSUSE: $> sudo zypper remove openss7 openss7-repo
SLES:     $> sudo zypper remove openss7 openss7-repo
Mageia:   $> sudo uprme openss7
Mandriva: $> sudo urpme openss7
MES:      $> sudo urpme openss7
Debian:   $> sudo aptitude remove openss7 openss7-repo
Ubuntu:   $> sudo aptitude remove openss7 openss7-repo
```

[2] Note that the apt-transport-https package must be installed for the openss7 repositories to function correctly. Try '`sudo aptitude install apt-transport-https`'.

Build Prerequisites

When building from source RPM or DSC, the prerequisites for building must be met. Most RPM or DEB build prerequisites are automatic; however, some prerequisites must still be met manually. When building from tarball, most prerequisites must be met manually. The `configure` script will inform you of most missing prerequisites and the actions that must be performed to meet those prerequisites.

Prerequisites for building OpenSS7 package are as follows:

1. *Linux* distribution, somewhat *Linux Standards Base* compliant, with a 2.4, 2.6 or 3.x 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 7.2 [Compatibility], page 62.
   - A fairly LSB compliant GNU/Linux distribution.[3]
   - Linux 2.4 kernel (2.4.10 - 2.4.27),
   - Linux 2.6 kernel (2.6.3 - 2.6.39), or
   - Linux 3.x kernel (3.0 - 3.14);
   - glibc2 or better.
   - GNU groff (for man pages).[4]
   - GNU texinfo (for info files).
   - GNU bison and flex (for config programs).
   - net-snmp (for SNMP agents).[5]
   - GNU gcj and classpath (for Java modules).
   - swig (for Java, Tcl, Perl and Ruby interfaces).

The following will meet most additional prerequisites for a CentOS/RHEL 5.5 build host:

```
#> yum install bzip2 chkconfig coreutils createrepo doxygen \
        gcc-java ghostscript gjdoc glibc gnupg gnuplot \
        groff gzip ImageMagick kernel-devel latex2html \
        libgcj lsof module-init-tools rpm rpm-build tetex \
        tetex-dvips tetex-latex transfig xz xz-lzma-compat \
        zip
```

The following will meet most additional prerequisites for a CentOS/RHEL 6.0 build host:

```
#> yum install bzip2 chkconfig coreutils createrepo doxygen \
gcc-java ghostscript glibc gnuplot groff gzip ImageMagick \
java-1.6.0-openjdk-devel java-1.6.0-openjdk-javadoc \
kabi-whitelists kernel kernel-devel latex2html libgcj \
libgcj-devel lsof module-init-tools net-snmp-devel \
perl-devel rpm rpm-build tcl-devel texlive texlive-dvips \
texlive-latex texlive-utils transfig xz zip
```

The following will meet most additional prerequisites for a SuSE Linux Enterprise 11 build host:

```
#> zypper install aaa_base bzip2 coreutils createrepo doxygen \
        fastjar gcc43-gij gcc-java ghostscript-library gjdoc \
        glibc gnuplot gpg2 groff gzip ImageMagick inst-source-utils \
        kernel-default-devel latex2html lsof module-init-tools \
        rpm texlive texlive-latex transfig zip
```

[3] See Section 7.2.1 [GNU/Linux Distributions], page 62, for more information.

[4] 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.

[5] A wide range of net-snmp releases are supported, from UCD-SNMP 4.2.5 through net-snmp 5.7.2.

The following will meet most additional prerequisites for a Debian 6.0 build host:

```
#> apt-get install apt-utils bzip2 coreutils createrepo doxygen \
        dpkg dpkg-dev fastjar gcj-jdk gcj-jre-headless ghostscript \
        gnupg gnuplot-nox gnuplot-x11 groff groff-base gzip \
        imagemagick insserv latex2html libc-bin lsof lzma \
        module-init-tools rpm texlive-binaries texlive-font-utils \
        texlive-latex-base transfig xz-utils zip
```

The package builds and installs kernel modules. When configuring and building the package, it is necessary to have the 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
CentOS:  $> 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:

```
$> ../openss7-1.1.7.20141001/configure CC='gcc-3.4'
```

The package builds and installs SNMP agents. When configuring and building the package, it is necessary to have the net-snmp development packages installed. For the following distributions, use the following commands:

```
Ubuntu:  $> apt-get install libsnmp libsnmp-perl snmp snmpd \
                 lm-sensors libsnmp-dev libsnmp9-dev libsensors-dev
Debian:  $> apt-get install libsnmp libsnmp-perl snmp snmpd \
                 lm-sensors libsnmp-dev libsnmp9-dev libsensors-dev
Fedora:  $> yum install net-snmp net-snmp-perl net-snmp-utils \
                 lm_sensors net-snmp-devel lm_sensors-devel
CentOS:  $> yum install net-snmp net-snmp-perl net-snmp-utils \
                 lm_sensors net-snmp-devel lm_sensors-devel
```

The package builds and installs Java archives and compiled Java. When configuring and building the package, it is necessary to have the GNU GCJ Java Compiler front-end and GNU Classpath archives installed. For the following distributions, use the following commands:

```
Ubuntu:  $> apt-get install gcj
Debian:  $> apt-get install gcj
Fedora:  $> yum install gcc-java
CendOS:  $> yum install gcc-java
```

Installation

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

```
$> wget http://www.openss7.org/tarballs/openss7-1.1.7.20141001.tar.xz
$> tar -xJvf openss7-1.1.7.20141001.tar.xz
$> mkdir build
$> pushd build
$> ../openss7-1.1.7.20141001/configure --enable-autotest --enable-silent-rules
$> make V=0
$> make check
$> sudo make install
$> sudo make installcheck
```

```
$> sudo make uninstall
$> popd
$> sudo rm -rf build
$> rm -rf openss7-1.1.7.20141001
$> rm -f openss7-1.1.7.20141001.tar.xz
```

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.[6] Installation steps using the logging targets proceed as follows:

```
$> wget http://www.openss7.org/tarballs/openss7-1.1.7.20141001.tar.xz
$> tar -xJvf openss7-1.1.7.20141001.tar.xz
$> mkdir build
$> pushd build
$> ../openss7-1.1.7.20141001/configure --enable-autotest --enable-silent-rules
$> make V=1 compile.log
$> make check.log
$> sudo make install.log
$> sudo make installcheck.log
$> sudo make uninstall.log
$> popd
$> sudo rm -rf build
$> rm -rf openss7-1.1.7.20141001
$> rm -f openss7-1.1.7.20141001.tar.xz
```

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

For custom applications, see the `INSTALL` and `INSTALL-openss7` files or the see Chapter 8 [Installation], page 109, as listed below. If you encounter troubles, see Chapter 9 [Troubleshooting], page 191, before issuing a bug report.

Brief Installation Instructions

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

```
$> wget http://www.openss7.org/tarballs/openss7-1.1.7.20141001.tar.xz
```

Unpack the tarball using a command such as:

```
$> tar -xJvf openss7-1.1.7.20141001.tar.xz
```

The tarball will unpack into the relative subdirectory named after the package name: openss7-1.1.7.20141001.

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
$> ../openss7-1.1.7.20141001/configure
```

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 ../openss7-1.1.7.20141001/INSTALL
```

For specific options to the `configure` script, see the `INSTALL-openss7` file in the distribution, or simply execute the configure script with the `--help` option like so:

[6] Please see Section 9.2 [Problem Reports], page 195, or the file `PROBLEMS` in the release directory for more information on filing a proper *Problem Report*.

```
$> ../openss7-1.1.7.20141001/configure --help
```

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

```
$> make V=0
```

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 `info` documentation is automatically installed; however, the text, html and pdf documentation must be installed separately like so:

```
$> sudo make install-txt
$> sudo make install-html
$> sudo make install-pdf
```

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 (including installed documentation) 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 openss7-1.1.7.20141001
$> rm -f openss7-1.1.7.20141001.tar.xz
```

Detailed Installation Instructions

More detailed installation instructions can be found in the Chapter 8 [Installation], page 109, contained in the distribution in '`text`', '`info`', '`html`' and '`pdf`' formats:

```
$> cd ../openss7-1.1.7.20141001
$> less doc/manual/openss7.txt
$> lynx doc/manual/openss7.html
$> info doc/manual/openss7.info
$> xpdf doc/manual/openss7.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/openss7_manual.html
```

1 Introduction

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

1.1 Overview

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

1.2 Organization of this Manual

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

Introduction.	This introduction
Objectives.	Objectives of the package
Reference.	Contents of the package
Development.	Developing with the package
Porting.	Porting to the package
Conformance.	Conformance of the package
Releases.	Releases of the package
Installation.	Installation of the package
Troubleshooting.	Troubleshooting of the package

1.3 Conventions and Definitions

This manual uses *texinfo* typographic conventions.

2 Objectives

2.1 Background

STREAMS derives from Dennis Ritchie's original paper,[1] was incorporated into the *UNIX® System V Release 3* operating system, replaced the terminal input-output subsystem, pipes and FIFOs in *UNIX® System V Release 4*, and was improved in the USL release of the *UNIX® System V Release 4.2* operating system.

Today, STREAMS is a part of every major branded *UNIX®* variant, such as *AIX®*, *HP-UX®*, *IRIX®*, *MacOT®*, *OSF/1®*, *Solaris®*, *SUPER-UX®*, *UnixWare®*, *UXP/V®*, and including many *UNIX*-like operating systems and popular embedded RTOS, but with the notable exception of *Berkeley System Distribution* releases, variants and offshoots, and *Linux*.

2.2 What is STREAMS?

STREAMS is a flexible framework for communication between a user level process and a kernel resident driver. It encompasses a set of kernel system calls providing a user-kernel interface that is backward compatible with the traditional character device driver interface, as well as a set of STREAMS driver and module entry points forming a driver-kernel interface. STREAMS also provides a rich set of kernel utility functions for the development and implementation of kernel-resident drivers and modules. STREAMS prompted the specification of the DDI/DKI which is an architecture independent driver-kernel interface that provides a standardized set of kernel functions (beyond just STREAMS) for the development of device and software drivers.

STREAMS provides a reconfigurable full-duplex communications path between user level process and kernel resident driver, termed a *Stream*. Modules can be inserted in the path between the user and driver under user level control. Streams can be linked across multiplexers under user control to form complex (yet reconfigurable) topologies of user level processes and drivers.

Communication of control and data information along a Stream is accomplished by message passing. There is no direct function call interface between components of a Stream. A Stream exists within the STREAMS framework inside the kernel and extend from the user-kernel interface to the kernel driver interface. Each component of a Stream consists of a pair of queues used to pass messages in the upstream direction to the kernel-user interface; or downstream, the kernel-driver interface.

At the kernel-user end of the Stream is a component called the *Stream head*. As with all components of a Stream, the Stream head consists of a queue pair and a specialized set of procedures. The Stream head procedures are responsible for converting between the system call interface presented to users and the message passing mechanism within the Stream.

At the kernel-driver end of the Stream is the *Stream end*. The Stream end also contains a queue pair and a set of procedures. The Stream end (or simply driver) procedures are responsible for converting between the message passing mechanism within the Stream and the actions and events of a hardware (or pseudo-) device.

Intermediate components within the Stream are called *Modules*. Modules consist of a queue pair for passing messages upstream and downstream, as well as a set of procedures for processing messages. Modules can be pushed onto the module stack between the Stream head and Stream end using a set of standardized input-output control commands.

In support of topologies more complex than these simple linear segments, STREAMS also provides a specialized Stream end (driver) called a *Multiplexing driver*. A Multiplexing driver has the ability

[1] "A Stream Input-Output System", *AT&T Bell Laboratories Technical Journal* **63**, No. 8 Part 2 (October, 1984), pp. 1897-1910.

to open multiple Streams to its upper interface (multiplexer) as well as linking multiple Streams beneath its lower interface (multiplexer). Again, a standardized set of input-output controls provide the user with the ability to configure a Multiplexing driver.

2.3 Why STREAMS?

With the ability to open multiple Streams to a driver, push and pop modules to and from the module stack on a Stream, and to link any Stream under a multiplexing driver–all under user control using standardized input-output controls–allows STREAMS to configure complex topologies to form protocol stacks.

Almost all specialized standard telecommunications software developed since 1990 was developed to run on STREAMS. This is for several reasons:

- Since 1990, STREAMS and the associated DDI/DKI has been, and remains, the only way to incorporate OEM protocol stacks into mainstream *UNIX*® system kernels.
- The original *UNIX System Laboratories* (later *X/Open* then later the *OpenGroup*) support for ITU-T developed OSI protocols, makes STREAMS amenable to an open model for development for ITU-T protocols. (ITU-T, formerly CCITT, is the *International Telecommunications Union – Telephone Sector* responsible for international telephone standards, and original developers of the OSI model.)

As a result, there is a significant body of commercial software implementing telecommunications protocol stacks that was developed, tested, validated, conformance tested, field verified, to run on STREAMS: and is still running on STREAMS.

The cost of reimplementation, retesting, revalidation, redoing conformance testing, and field re-verification, would likely be prohibitive: after all, the point of *Linux* is reducing cost, is it not?

2.4 Why STREAMS for Linux?

The *Linux* kernel was not developed with STREAMS in mind. For TPI/IP networking, *Linux* originally followed in the footsteps of the BSD NET2 release. Currently, the implementation of TCP/IP in the *Linux* kernel has long departed from the classical BSD organization and exhibits characteristics unique to the *GNU/Linux* operating system. For character device and terminal input-output, *Linux* follows closely the SVR 3 pre-STREAMS approach to pipes, FIFOs and terminal subsystem. The terminal subsystem implementation, too, has become unique to *GNU/Linux*.

Therefore, from the perspective of TCP/IP networking and Terminal I/O, there would be little reason to provide STREAMS for *Linux*. That is, if it were not for the body of software supporting OSI and telecommunications protocols based solely on STREAMS, for which *Linux* has little or no support.

So, the answer to the question, "Why STREAMS for Linux?" is: so that a *GNU/Linux* platform can enjoy the same wealth of telecommunications and OSI protocol stacks otherwise only available to big-iron *UNIX*®. Without STREAMS, *Linux* is probably just another BSD, and probably not a very good one.

2.5 History of STREAMS for Linux

In 2000, The OpenSS7 Project abandoned using the *Linux* networking model for implementation of the *Signalling System No. 7* protocol (primarily due to the lack of support for the full BSD networking model under *Linux*) and switched to using STREAMS as the basis for all future development. Over the span of the next 5 years, (and not surprisingly given the body of software), almost all *Signalling System No. 7* products released on *Linux* used STREAMS.

In 2005, The OpenSS7 Project release (after two years of development) the `streams-0.7a.4` package: a reimplementation of *SVR 4.2 STREAMS* with compatibility modules for all major *UNIX*® releases, called *OpenSS7*. *OpenSS7* was intended as a *POSIX/SUSv3 XSR* conforming, high performance, production grade, implementation, suitable for mainline *Linux* adoption, and a better foundation on which to base SIGTRAN, VoIP, ISDN and SS7 protocol stacks developed under the The OpenSS7 Project, as well as a better foundation for porting commercial *UNIX*® OEM implementations to *Linux*. It is the `openss7-1.1.7.20141001` package that contains the documentation you are reading now.

2.6 Why Fast?

In late 2003, The OpenSS7 Project decided to begin implementation of a production grade implementation of STREAMS, because of a number of shortcomings of other *Linux* STREAMS implementations:

a. proprietary;
b. unsuitable for mainline kernel adoption due to coding style or organization;
c. poorly adapted to distribution production kernels;
d. is unsuitable for packaging or repeatability;
e. portability objective unsuitable for mainline kernel adoption;
f. ports from the same baseline obfuscate the code;
g. poor performance due to portability or coding style;
h. code bloat or over sized memory footprint;
i. redundant debug statements obscuring defects or obfuscating code;
j. overuse of semaphores;
k. contain serious races or not suitable for threaded applications;
l. non-conformance to mainstream *UNIX*® implementations;
m. non-conformance to *POSIX* or any release of the *Single UNIX Specification*;
n. limited set of standard drivers or modules;
o. limited set of diagnostic or administrative utilities;
p. limited test programs;
q. poorly documented.

The replacement, named *OpenSS7*, was to correct all of these difficulties, and, by the initial '`streams-0.7a.4`' release, was:

a. open source;
b. completely Lindented and follows kernel coding practises;
c. automatically adapts to production kernels with autoconf;
d. packages itself into LSB compliant RPMs and DEBs;
e. designed and implemented specifically for *GNU/Linux*;
f. no ports considered;
g. over twice the performance;
h. less than one-eighth of the memory footprint;
i. proper programming by assertion;
j. proper use of lightweight spin locks;

k. race free locking strategies and synchronization;
l. compatible with all mainstream *UNIX*® implementations;
m. conforms to *POSIX/SUSv3 XSR*;
n. complete set of standard drivers and modules;
o. complete set of diagnostic and administrative utilities;
p. integrated set of conformance test suites;
q. fully documented.

3 Reference

3.1 Files

The following kernel modules are installed by *OpenSS7* in the `/lib/modules/3.0.99-1-unx/openss7/`directory, with either a '`.o`' or '`.ko`' extension.[1]

`specfs`
: This kernel module contains the STREAMS Special Shadow Filesystem. See `specfs(5)` for more information.

`streams`
: This kernel module contains the STREAMS scheduler, utility functions, and STREAMS Device Driver Interface/Driver Kernel Interface (DDI/DKI). See `STREAMS(9)` for more information.

`streams-fifo`
: This kernel module contains the `fifo` STREAMS driver. This is a standard STREAMS driver, but is also used by the conformance and validation test suite. See `fifo(4s)` for more information.

`streams-sad`
: This kernel module contains the `sad` STREAMS driver. This is the standard STREAMS Administrative Driver. See `sad(4)` for more information.

`streams-nsdev`
: This kernel module contains the `nsdev` STREAMS driver. This is a *OpenSS7* specific driver. See `nsdev(4)` for more information.

`streams-echo`
: This kernel module contains the `echo` STREAMS driver. This is a standard STREAMS driver, but is also used by the conformance and validation test suite. See `echo(4)` for more information.

`streams-mux`
: This kernel module contains the `mux` STREAMS driver. This is a standard STREAMS driver< but is also used by the conformance and validation test suite. See `mux(4)` for more information.

`streams-nuls`
: This kernel module contains the `nuls` STREAMS driver. This is a standard STREAMS module. See `nuls(4)` for more information.

`streams-pipe`
: This kernel module contains the `pipe` STREAMS driver. This is a standard STREAMS driver. See `pipe(4)` for more information.

`streams-log`
: This kernel module contains the `log` STREAMS driver. This is a standard STREAMS driver. See `log(4)` for more information.

[1] The kernel version, 3.0.99-1-unx is just an example. When installed with a '`.ko`' extension, dashes ('`-`') are also changed to underscores ('`_`').

`streams-loop`
: This kernel module contains the `loop` STREAMS driver. This is a standard STREAMS driver, but is also used by the conformance and validation test suite. See `loop(4)` for more information.

`streams-sfx`
: This kernel module contains the *sfx* STREAMS driver. This is a common character device driver for implementing STREAMS FIFOs. See `sfx(4)` for more information.

`streams-spx`
: This kernel module contains the *spx* STREAMS driver. This is a common character device driver for implementing STREAMS pipes. See `spx(4)` for more information.

`streams-srvmod`
: This kernel module contains the `srvmod` STREAMS module. The `srvmod` STREAMS module is a simple buffer module (a module that always defers to its service procedure and then passes any message along). This module is used for performance testing of the STREAMS package. See `srvmod(4)` for more information.

`streams-nullmod`
: This kernel module contains the `nullmod` STREAMS module. The `nullmod` STREAMS module is a simple null module (a module that always passes messages to the next module in along the Stream). This module is used for performance testing of the STREAMS package and is also used by the conformance and validation test suite. See `nullmod(4)` for more information.

`streams-pipemod`
: This kernel module contains the `pipemod` STREAMS module. This is a standard STREAMS module used with pipes. See `pipemod(4)` for more information.

`streams-connld`
: This kernel module contains the `connld` STREAMS module. This is a standard STREAMS module. See `connld(4)` for more information.

`streams-sc`
: This kernel module contains the *sc* STREAMS module. This is a common STREAMS Configuration module. See `sc(4)` for more information.

`streams-testmod`
: This kernel module contains the `testmod` STREAMS module. This is a *OpenSS7* specific test module that is used for conformance and validation testing of STREAMS. See `testmod(4)` for more information.

Additional kernel modules are provided by add-on packages.

3.2 Drivers

The configuration of STREAMS drivers and modules is performed when compiling the *OpenSS7* subsystem. The STREAMS subsystem, core drivers and modules are part of every *OpenSS7* system. The following lists the core drivers and modules, STREAMS kernel tunable parameters, and STREAMS configuration information:[2]

[2] Note that documentation for STREAMS drivers is in the form of manual pages in Section 4 of the manual pages with the same name as the driver.

`clone(4)` (`streams`)
: Clone device driver. This is a standard *SVR 4.2 STREAMS* driver. The `clone(4)` driver is a integral part of STREAMS and is used to create clone instances of a STREAMS driver.

  See `clone(4)` for more information.

`echo(4)` (`streams-echo`)
: Echo (loopback) device driver. This is a commonly implemented STREAMS driver. It is implemented by *HP-UX*® and *OSF/1*®. The `echo(4)` driver provides a simple FIFO-like device without full *POSIX* FIFO semantics. Its primary purpose is for the STREAMS Verification function, `strvf(8)`, and the `test-streams(8)` validation test suite.

  See `echo(4)` for more information.

`fifo(4s)` (`streams-fifo`)
: FIFO (Named Pipe) device driver. This is a standard *SVR 4.2 STREAMS* driver. The `fifo(4s)` driver provides *POSIX*-compliant STREAMS-based *FIFO* device. Not all implementations of STREAMS provide STREAMS-based *FIFO*s: some implementations use the older *SVR 3*-style *FIFO*s that are not STREAMS-based. *OpenSS7* provides STREAMS-based *FIFO*s with the `fifo(4s)` driver.

  See `fifo(4s)` for more information.

`log(4)` (`streams-log`)
: STREAMS log driver. This is a standard *SVR 4.2 STREAMS* driver. The `log(4)` driver provides a STREAMS capable logger in addition to the *BSD* logger present in *Linux*. The `log(4)` driver provides additional support for STREAMS modules and drivers using the `strlog(9)` kernel level utility. *OpenSS7* also provides the `strace(8)`, `strerr(8)` and `strclean(8)` administrative utility functions and startup scripts for controlling the `log(4)` driver.

  See `log(4)` for more information.

`loop(4)` (`streams-loop`)
: Loop device driver. This is a standard *SVR 4.2 STREAMS* driver. The `loop` driver is detailed in the *UNIX System V Release 4 Programmer's Manual – STREAMS*. The `loop(4)` driver provides capabilities used primarily for validation test programs (see `test-streams(8)`) as well as serving as an example driver.

  See `loop(4)` for more information.

`mux(4)` (`streams-mux`)
: Multiplexing driver. This is a standard *SVR 4.2 STREAMS* driver. The `mux` driver is detailed in the *UNIX System V Release 4 Programmer's Manual – STREAMS*. The `mux(4)` driver provides capabilities used primarily for validation test programs (see `test-streams(8)` as well as serving as an example multiplexing driver. This `mux(4)` driver also provides the `minimux` capabilities.

  See `mux(4)` for more information.

`nsdev(4)` (`streams-nsdev`)
: Named STREAMS device driver. This is a *OpenSS7* specific driver. The `nsdev(4)` driver is a `clone(4)`-like driver that permits the specification of major and minor device numbers using the device node name. It provides one of three mechanisms under *OpenSS7* that remove STREAMS driver dependency on statically allocated device numbers.

See nsdev(4) for more information.

nuls(4) (streams-nuls)
: Null *Stream* driver. This is a standard *SVR 4.2 STREAMS* driver. The nuls(4) driver is usually called '`null`'. *Linux* has its own *SVR3*-style `/dev/null` driver, so it was renamed to '`nuls`'.

  See nuls(4) for more information.

pipe(4) (streams-pipe)
: STREAMS-based pipe driver. This is a standard *SVR 4.2 STREAMS* driver. However, pipe(4) is not normally implemented as a STREAMS driver, but is implemented as a system call. *OpenSS7* provides pipe(2s) system call emulation which invokes this driver internal to the kernel.

  See pipe(4) for more information.

sad(4) (streams-sad)
: STREAMS Administrative Driver. This is a standard *SVR 4.2 STREAMS* driver. The sad(4) driver is used by the autopush(8) utility to examine and specify the autopush lists for STREAMS drivers. Also, it is used to examine and verify the present of STREAMS modules or drivers in the system.

  See sad(4) for more information.

sfx(4) (streams-sfx)
: STREAMS FIFO device driver. This is commonly implemented STREAMS driver that is used to implement STREAMS FIFOs (Named Pipes) using a regular character device. The sfx(4) driver provides a character based device approach to creating FIFOs.

  See sfx(4) for more information.

spx(4) (streams-spx)
: STREAMS pipe device driver. This is commonly implemented STREAMS driver that is used to implement STREAMS pipes using a regular character device. The spx(4) driver provides a character based device approach to creating FIFOs and pipes. Only *UnixWare*® and *AIX(4)* document this device.

  See spx(4) for more information.

Additional drivers are provided by add-on packages.

3.3 Modules

The configuration of STREAMS drivers and modules is performed when compiling the *OpenSS7* subsystem. The STREAMS subsystem, core drivers and modules are part of every *OpenSS7* system. The following lists the core drivers and modules, STREAMS kernel tunable parameters, and STREAMS configuration information:[1]

`pipemod(4)` (`streams-pipemod`)
: Pipe module. This is a standard *SVR 4.2 STREAMS* module. The `pipemod(4)` module can be pushed over a pipe end or FIFO before other modules are pushed (on either end) to reverse the sense of the `M_FLUSH(9)` message that traverse the pipe.

  See `pipemod(4)` for more information.

`connld(4)` (`streams-connld`)
: Connection Line Discipline module. This is a standard *SVR 4.2 STREAMS* module. The `connld(4)` module can be pushed over a pipe end that has been attached to a file system file using `fattach(3)` and will then create a new pipe instance on each `open(2s)` of the attached file and pass the new remove file pointer to the remove end using `M_PASSFP(9)` to be received with `I_RECVFD(7)`. This allows servers to be created that use `pipe(4)`s for communication.

  See `connld(4)` for more information.

`sc(4)` (`streams-sc`)
: STREAMS Configuration module. This is a commonly implemented STREAMS module. It is implemented by *HP-UX* and *AIX*, and perhaps other *Mentat*-derived STREAMS implementations. The `sc(4)` modules provides the ability to access STREAMS driver information by name rather than major device number. It also provides access to the `module_info(9)` and `module_stat(9)` structure information for the named STREAMS module or driver, not accessible using the `sad(4)` driver. The `sc(4)` module is used by the `scls(8)` utility.

  See `sc(4)` for more information.

`srvmod(4)` (`streams-srvmod`)
: Buffer module. This is a standard *SVR 4.2 STREAMS* module described in the *UNIX System V Release 4 Programmer's Manual – STREAMS*. The `srvmod(4)` module also has *OpenSS7* specific extensions. The `srvmod(4)` module is used by the `perftest(8)` performance test program to test the effect of additional levels of service procedure pushed over a *Stream*. The module also serves as an example of a STREAMS module using service procedures.

  See `srvmod(4)` for more information.

`nullmod(4)` (`streams-nullmod`)
: Null module. This is a standard *SVR 4.2 STREAMS* module described in the *UNIX System V Release 4 Programmer's Manual – STREAMS*. The `nullmod(4)` module also has *OpenSS7* specific extensions. The `nullmod(4)` module is used by the `perftest(8)` performance test program to test the effect of additional levels of put procedure pushed over a *Stream*. The module also serves as an example of a STREAMS module not using service procedures.

  See `nullmod(4)` for more information.

[1] Note that documentation for STREAMS modules is in the form of a manual page in Section 4 of the manual pages with the same name as the module.

`testmod(4)` (`streams-testmod`)
: Test module. This is a *OpenSS7* specific STREAMS module. The primary purpose of the `testmod(4)` modules is to provide the `test-streams(8)` validation test program with the capability to pass specific `M_ERROR(9)` and `M_HANGUP(9)` messages to the *Stream head* for *POSIX* validation testing. It also serves as an example of how a STREAMS module can properly process `M_IOCTL(9)` and related messages.

  See `testmod(4)` for more information.

Additional modules are provided by add-on packages.

3.4 Libraries

During the installation process of *OpenSS7* a subroutine library is built and installed on your system. For 64-bit systems that support 32-bit compatibility, two versions of each library are built and installed: one 64-bit native library and one 32-bit compatibility library. 64-bit native libraries are installed to the `/usr/lib64` subdirectory. 32-bit native and 32-bit compatibility libraries are installed to the `/usr/lib` subdirectory.

`libstreams.so.0.0.1`
`libstreams.so.0`
`libstreams.so`
: Provides a shared object library for use by STREAMS applications programs.

`libstreams.a`
: Provides a static library for use by STREAMS applications programs.

`libstreams.la`
: Provides the `libtool` definitions for the library.

3.4.1 libstreams Library Routines

The following routines are present in the `libstreams` libraries. The routines in these libraries are standard STREAMS interface system calls documented in the *System V Release 4.2 Programmer's Manual – STREAMS*. Refer to the associated manual pages for detailed information on these routines.

`fattach(2)`
: Name a STREAMS special file.

`fdetach(2s)`
: Unname a STREAMS special file.

`getmsg(2s)`
: Get next message off of a Stream.

`getpmsg(2s)`
: Get next message off of a Stream.

`isastream(2s)`
: Test for a STREAMS special file.

`pipe(2s)` Create a STREAMS pipe.

`putmsg(2s)`
: Put a message to a STREAMS character device.

`putpmsg(2s)`
: Put a band message to a STREAMS character device.

`pstrlog(3)`
: Print a STREAMS log buffer.

`strlog(3)` Print a STREAMS log buffer.

`vstrlog(3)`
: Print a STREAMS log buffer.

3.4.2 Using the Library

To use one of the *OpenSS7* libraries you can include the file `sys/stropts.h` in you application program source code. On you compiler command line, add the option '`-I/usr/include/openss7`' to include the version of `sys/stropts.h` that is distributed with *OpenSS7*.

When linking our program, or performing a final `gcc` to build your executable, include one of the following options on your command line:

'`/usr/lib/libstreams.a`'
'`-lstreams -static`'
: Link against the static version of the library.

'`-lstreams`'
: Link against the shared object version of the library.

'`/usr/lib/libstreams.la`'
: Use with `libtool` to link additional convenience libraries against the shared or static versions of the library.

Failure to link the executable runtime path for `libstreams` will result in linker-loader warnings that the functions `getpmsg(2s)` or `putpmsg(2s)` are not implemented and will always fail.[1].

See also Chapter 4 [Development], page 39 for more information.

[1] These warnings are generated when linking only against the `libc` library that includes only failing stubs for `getpmsg(2s)` and `putpmsg(2s)`

3.5 Utilities

3.5.1 Init Scripts

Following are System V Init Scripts that are installed by the package:

`specfs(8)` (`/etc/init.d/specfs`)
`specfs.sh(8)` (`/etc/init.d/specfs.sh`)
: System V Init Script for the STREAMS Special Shadow Filesystem. The `specfs(8)` init script provides the ability to initialize, configure and mount the STREAMS Special Shadow Filesystem, `specfs(5)`. The `specfs(8)` script provides the RedHat-style init script, whereas the `specfs.sh(8)` script provides the Debian-style init script.

  See `specfs(8)` for more information.

`openss7(8)` (`/etc/init.d/openss7`)
`openss7.sh(8)` (`/etc/init.d/openss7.sh`)
: System V Init Script for the STREAMS Subsystem. The `streams(8)` init script provides the ability to initialize, configure and mount the STREAMS subsystem, `STREAMS(9)`. The `streams(8)` script provides the RedHat-style init script, whereas the `streams.sh(8)` script provides the Debian-style init script.

  See `streams(8)` for more information.

3.5.2 User Utilities

Following are user utilities for manipulating *Streams*:

`strchg(1)` (`/usr/bin/strchg`)
: Change *Stream* configuration. `strchg(1)` is a standard *SVR 4.2 STREAMS* user utility.

  `strchg(1)` is a C-language user program that can be used to alter the configuration of the *Stream* associated with the caller's standard input. The `strchg(1)` command pushes modules on the *Stream*, pops modules off of the *Stream*, or both. Only the superuser or owner of the STREAMS device can alter the configuration of that *Stream*. If another user attempts to alter the configuration, the `strconf(1)` command will fail.

  `strchg(1)` is useful from the shell and, when standard input is redirected from an open file descriptor to the command, can be used to push and pop modules from arbitrary *Streams*, not just those associated with STREAMS-based terminal devices.

  See `strchg(1)` for more information.

`strconf(1)` (`/usr/bin/strconf`)
: Query *Stream* configuration. `strconf(1)` is a standard *SVR 4.2 STREAMS* user utility.

  `strconf(1)` is a C-language user program that can be used to query the configuration of a *Stream*. When use without any options, it prints a list of the modules in the *Stream* associated with the standard input, as well as the topmost driver. The list is printed with one name per line, where the first name printed is the topmost module on the *Stream* and the last item printed is the name of the topmost driver associated with the *Stream*.

  `strconf(1)` is useful from the shell and, when standard input is redirected from an open file descriptor to the command, can be used to query arbitrary *Streams*, not just the associated with STREAMS-based terminal devices.

  See `strconf(1)` for more information.

`strreset(1)` (`/usr/bin/strreset`)
: Reset a *Stream*. `strreset(1)` is a standard *SVR 4.2 STREAMS* user utility.

  `strreset(1)`is a C-language user program that resets an open *Stream* by generating an `M_FLUSH(9)` message to the *Stream head*. It is used mainly to reset blocked *Streams*. Wehn it is impossible to reopen the *Stream*, issue an `I_FLUSH` or equivalent command. This situation may happen with a process sleeping in a module's close routine, when signals can not be sent to the process (a zombie process exiting, for example).

  See `strreset(1)` for more information.

3.5.3 Administrative Utilities

Following are administrative utilities for manipulating and examining the STREAMS subsystem:

`autopush(8)` (`/usr/sbin/autopush`)
: Control the *autopush* module list for a STREAMS device. `autopush(8)` is a standard *SVR 4.2 STREAMS* administrative utility.

  `autopush(8)` is a C-language program that can be used to manipulate and examine which STREAMS modules are automatically pushed over a device when it is opened. It is also possible to restrict the ability to push further modules on the *Stream* without proper privilege. The `autopush(8)` utility provides a user program interface to the STREAMS Administrative Driver (`sad(4)`).

  See `autopush(8)` for more information.

`fattach(8)` (`/usr/sbin/fattach`)
: Name a STREAMS file. `fattach(8)` is an *OpenSS7* utility. Although *OSF/1* documentation mentions an `fattach` manual page in section 8, one does not exist.

  `fattach(8)` opens a `pipe(4)` and attaches one end of the pipe to a file using `fattach(3)`, and optionally pushes the `connld(4)` module on the side of the pipe being attached to the file. The other end of the pipe remains available for use by the shell program invoking this command.

  `fattach(8)` provides a easy way for shell programs to use STREAMS-based pipes and to use the facilities of the `connld(4)` module.

  See `fattach(8)` for more information.

`fdetach(8)` (`/usr/sbin/fdetach`)
: Unlink a named STREAMS file.

  `fdetach(8)` is a standard *SVR 4.2 STREAMS* administrative utility.

  `fdetach(8)` is a C-language program that detaches or disassociates a file descriptor for an open STREAMS device or pipe from its filename in the file system.

  See `fdetach(8)` for more information.

`insf(8)` (`/usr/sbin/insf`)
: Install special files. `insf(8)` is the *HP-UX* way to install special (device) files. This program is not even partially implemented in *OpenSS7*. Use `streams_mknod(8)` and friends instead.

  See `insf(8)` for more information.

`scls(8)` (`/usr/sbin/scls`)
: List STREAMS configuration. `scls(8)` is a rather useful *AIX* administrative utility that is also implemented by *OpenSS7*.

`scls(8)` is a C-language program that can be used to list module and driver names as well as information and statistics associated with those modules or drivers. The `scls(8)` utility provides a user program interface to the STREAMS Configuration module (`sc(4)`).

See `scls(8)` for more information.

`strace(8)` (`/usr/sbin/strace`)
: Write STREAMS event trace messages to the standard output. `strace(8)` is a standard *SVR 4.2 STREAMS* administrative utility.

  The `strace(8)` C-language program receives trace event messages from the STREAMS log driver (`log(4)`) and writes these messages to the standard output. When run as a daemon, `strace(8)` appends these messages to a log file.

  Messages that appear in the trace log are intended to report debugging information that assists with troubleshooting a running STREAMS module or driver.

  See `strace(8)` for more information.

`strclean(8)` (`/usr/sbin/strclean`)
: Clean up after the STREAMS error logger. `strclean(8)` is a standard *SVR 4.2 STREAMS* administrative utility.

  The `strclean(8)` utility is a bash script that can be used to delete aged log files generated by the STREAMS error logger, `strerr(8)`.

  See `strclean(8)` for more information.

`streams_mknod(8)` (`/usr/sbin/streams_mknod`)
: Make special device nodes for STREAMS. `streams_mknod(8)` is a *OpenSS7* specific administrative utility.

  The `streams_mknod(8)` C-language program can be used to make (or remove) the special device nodes under the `/dev` directory required by `openss7-1.1.7.20141001` package modules and drivers. `streams_mknod(8)` in invoked by the System V startup script, `/etc/init.d/openss7`.

  See `streams_mknod(8)` for more information.

`strerr(8)` (`/usr/sbin/strerr`)
: Receive error log messages from the STREAMS `log(4)` driver. `strerr(8)` is a standard *SVR 4.2 STREAMS* administrative utility.

  The `strerr(8)` utility is a C-language program, run as a daemon, that receives error log messages from the STREAMS log driver (`log(4)`) and writes these message to a log file. By default, `strerr(8)` logs all *STREAM* error messages from all drivers and modules.

  Messages that appear in the error log are intended to report exceptional conditions that require the attention of the person who administers your system.

  See `strerr(8)` for more information.

`strinfo(8)` (`/usr/sbin/strinfo`)
: List *Stream* information. `strinfo(8)` is a rather useful *AIX* administrative utility that is also implemented by *OpenSS7*.

  The `strinfo(8)` C-language program can be used to list *Stream* instance information as well as information and statistics on a module or driver basis. The `scls(8)` utility provides a user program interface to the STREAMS Configuration module (`sc(4)`).

This program is not even partially implemented in *OpenSS7* yet. User `proc(5)` file system and the `/proc/streams` directory instead. Also, see `scls(8)` for driver and module specific information.

See `strinfo(8)` for more information.

`strload(8)` (`/usr/sbin/strload`)
Loads the STREAMS subsystem. `strload(8)` is a useful *AIX* administrative utility that is also implemented by *OpenSS7*.

The `strload(8)` bash script can be used to load STREAMS modules and drivers individually or from a configuration file.

See `strload(8)` for more information.

`strsetup(8)` (`/usr/sbin/strsetup`)
Bash script.

See `strsetup(8)` for more information.

`strvf(8)` (`/usr/sbin/strvf`)
C-language program.

See `strvf(8)` for more information.

3.5.4 Performance Test Programs

Following are performance test programs:

`perftest(8)` (`/usr/sbin/perftest`)
C-language program.

See `perftest(8)` for more information.

`perftestn(8)` (`/usr/sbin/perftestn`)
C-language program.

See `perftestn(8)` for more information.

3.5.5 Conformance Test Programs

Following and conformance and validation testing programs:

`test-clone(8)` (`/usr/libexec/openss7/test-clone`)
The `test-clone(8)` C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the `clone(4)` STREAMS driver.

See `test-clone(8)` for more information.

`test-connld(8)` (`/usr/libexec/openss7/test-connld`)
The `test-connld(8)` C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the `connld(4)` STREAMS driver.

See `test-connld(8)` for more information.

`test-echo(8)` (`/usr/libexec/openss7/test-echo`)
The `test-echo(8)` C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the `echo(4)` STREAMS driver.

See `test-echo(8)` for more information.

`test-fifo(8)` (`/usr/libexec/openss7/test-fifo`)
The `test-fifo(8)` C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the `fifo(4s)` STREAMS driver.

See test-fifo(8) for more information.

test-log(8) (/usr/libexec/openss7/test-log)
: The test-log(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the log(4) STREAMS driver.

  See test-log(8) for more information.

test-loop(8) (/usr/libexec/openss7/test-loop)
: The test-loop(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the loop(4) STREAMS driver.

  See test-loop(8) for more information.

test-mux(8) (/usr/libexec/openss7/test-mux)
: The test-mux(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the mux(4) STREAMS driver.

  See test-mux(8) for more information.

test-nsdev(8) (/usr/libexec/openss7/test-nsdev)
: The test-nsdev(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the nsdev(4) STREAMS driver.

  See test-nsdev(8) for more information.

test-nuls(8) (/usr/libexec/openss7/test-nuls)
: The test-nuls(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the nuls(4) STREAMS driver.

  See test-nuls(8) for more information.

test-pipe(8) (/usr/libexec/openss7/test-pipe)
: The test-pipe(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the pipe(4) STREAMS driver.

  See test-pipe(8) for more information.

test-pipemod(8) (/usr/libexec/openss7/test-pipemod)
: The test-pipemod(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the pipemod(4) STREAMS driver.

  See test-pipemod(8) for more information.

test-sad(8) (/usr/libexec/openss7/test-sad)
: The test-sad(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the sad(4) STREAMS driver.

  See test-sad(8) for more information.

test-sc(8) (/usr/libexec/openss7/test-sc)
: The test-sc(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the sc(4) STREAMS driver.

  See test-sc(8) for more information.

test-streams(8) (/usr/libexec/openss7/test-streams)
: The test-streams(8) C-language program is a conformance and validation test program, in the OpenSS7 Project style, for the STREAMS(9) subsystem and primarily the sth(4) *Stream head*.

  See test-streams(8) for more information.

For the proper way to execute these validation test programs in a conformance and validation test suite, see Section 9.1.2.1 [Running Test Suites], page 194.

4 Development

For development using the *OpenSS7* package, See Section "About This Manual" in *STREAMS Programmer's Guide*.

4.1 Header Files

Header files are installed, typically, in the `/usr/include/openss7` subdirectory. To use the header files from the package, '`-I/usr/include/openss7`' must be included in the `gcc` command line as a compile option. This is true regardless of whether user space or kernel space programs are being compiled.

In general, '`-I`' include directives on the `gcc` command line should be ordered in the reverse order of the dependencies between packages. So, for example, if the include files from all add-on packages are required, the order of these directives would be: '`-I/usr/include/strss7 -I/usr/include/strsctp -I/usr/include/strinet -I/usr/include/strxnet -I/usr/include/strxns -I/usr/include/strcompat -I/usr/include/streams`'.

Following are the user visible header files provided by the `openss7-1.1.7.20141001` package in directory `/usr/include/streams`:

`strlog.h`
: This is the primary header file for the `strlog(4)` driver. It is normally only included by user space programs when interacting with the `log(4)` driver. See `log(4)` for more information.

`stropts.h`
: This is the primary user header file for the *Stream head*. It is normally only included by user space programs when interacting with the *Stream head*. See `sth(4)` for more information.

`log.h`
: This is the primary header file for the `log(4)` driver. It is normally only included by user space programs when interacting with the `log(4)` driver. See `log(4)` for more information.

`loop.h`
: This is the primary header file for the `loop(4)` driver. It is normally only included by user space programs when interacting with the `loop(4)` driver. See `loop(4)` for more information.

`sad.h`
: This is the primary header file for the `sad(4)` driver. It is normally only included by user space programs when interacting with the `sad(4)` driver. See `sad(4)` for more information.

`sys/cmn_err.h`
: This is the system specific kernel header file for the `cmn_err(9)` utility.

`sys/ddi.h`
: This is the system specific kernel header file for various STREAMS `DDI(9)` utilities. It is normal only included by kernel space STREAMS modules and drivers. See `DDI(9)` for more information.

`sys/debug.h`
: This is the system specific kernel header file for kernel debugging macros. It is normal only included by kernel space STREAMS modules and drivers.

`sys/dki.h`
: This is the system specific kernel header file for various STREAMS `DKI(9)` utilities. It is normal only included by kernel space STREAMS modules and drivers. See `DKI(9)` for more information.

`sys/kmem.h`
: This is the system specific kernel header file for `kmem_alloc(9)` and related utilities. It is normal only included by kernel space STREAMS modules and drivers. See `kmem_alloc(9)` for more information.

`sys/strconf.h`
: This is the system specific kernel header file for STREAMS driver and module configuration. It is normal only included by kernel space STREAMS modules and drivers.

`sys/strdebug.h`
: This is the system specific kernel header file for STREAMS driver and module debugging macros. It is normal only included by kernel space STREAMS modules and drivers.

`sys/stream.h`
: This is the system specific kernel header file for STREAMS drivers and modules. It is normal only included by kernel space STREAMS modules and drivers. See `STREAMS(9)` for more information.

`sys/strlog.h`
: This is the system specific header file for the `strlog(4)` and `strlog(9)` facilities. It is normally only included by kernel space programs when interacting with the `log(4)` driver. See `log(4)` for more information.

`sys/stropts.h`
: This is the system specific user header file for the *Stream head*. It is normally only included by user space programs when interacting with the *Stream head*. See `sth(4)` for more information.

`sys/stropts32.h`
: This is the system specific user 32/64-bit header file for the *Stream head*. It is normally only included by user space programs when interacting with the *Stream head*. See `sth(4)` for more information.

`sys/strsubr.h`
: This is the system specific kernel header file for STREAMS private definitions. It is normal only included by kernel space STREAMS modules and drivers. See `STREAMS(9)` for more information.

`sys/log.h`
: This is the system specific header file for the `log(4)` driver. It is normally only included by kernel space programs when interacting with the `log(4)` driver. See `log(4)` for more information.

`sys/loop.h`
: This is the system specific header file for the `loop(4)` driver. It is normally only included by kernel space programs when interacting with the `loop(4)` driver. See `loop(4)` for more information.

`sys/sad.h`

This is the system specific header file for the `sad(4)` driver. It is normally only included by kernel space programs when interacting with the `sad(4)` driver. See `sad(4)` for more information.

`sys/sc.h`

This is the system specific header file for the `sc(4)` module. It is normally only included by user or kernel space programs when interacting with the `sc(4)` driver. See `sc(4)` for more information.

`sys/testmod.h`

This is the system specific header file for the `testmod(4)` module. It is normally only included by user or kernel space programs when interacting with the `testmod(4)` driver. See `testmod(4)` for more information.

4.1.1 User Space Programs

Typical include files for interacting with STREAMS from user space include the `stropts.h` header file. Additional header files for interacting with specific drivers or modules may also be required.

4.1.2 Kernel Space Drivers and Modules

Typical include files for writing STREAMS modules and drivers for kernel space include the `sys/cmn_err.h`, `sys/kmem.h`, `sys/dki.h`, `sys/stream.h`, `sys/ddi.h`, and `sys/strconf.h` header files. Additional header files for interacting with specific drivers or modules may also be required.

4.2 Libraries

Shared or static versions of the `libstreams` library must be linked when using the `openss7-1.1.7.20141001` package. This library must either be specified on the `gcc` command line as a shared library (e.g. '`-lstreams`') or as a static library (e.g. '`/usr/lib/libstreams.a`').

If the shared library is linked, include the following options on the `gcc` command line:

'`-lstreams`'

Link to the `/usr/lib/libstreams.so` shared library.

If the static library is linked, include the following options on the `gcc` command line:

'`/usr/lib/libstreams.a`'

Link to the `/usr/lib/libstreams.a` static library.

4.3 Kernel Modules

Developing STREAMS kernel modules is similar to user space programs with regard to header files. `/usr/include/openss7` should be placed as an include directory to search in the `gcc` command line. The rules for compiling *Linux* kernel modules should be followed. In particular, several important intricacies should be considered:

- The `gcc` compiler used to compile the kernel modules must be the same version of compiler that was used to compile the kernel.
- The `gcc` command line must have the same compile flags that were used to compile the kernel.
- The `gcc` command line must define several important kernel defines including '`-DLINUX`', '`-D__KERNEL__`', as well as the base name of the module.

- The `gcc` command line must include several important include files directly on the command line such as '`--include /lib/modules/3.0.99-1-unx/build/include/linux/autoconf.h`' and maybe even '`--include /lib/modules/3.0.99-1-unx/build/include/linux/modversions.h`'.[1]

4.4 Manual Pages

The `openss7-1.1.7.20141001` package installs a number of manual pages in the `/usr/share/man` directory as follows:

The following manual pages are installed in Section 1 of the manual (in the subdirectory `/usr/share/man/man1`):

`strchg(1)` –	change Stream configuration.
`strconf(1)` –	query Stream configuration.
`strreset(1)` –	reset a Stream.

The following manual pages are installed in Section 2 of the manual (in the subdirectory `/usr/share/man/man2`):

`fattach(2)` –	name a STREAMS special file.
`fdetach(2s)` –	unname a STREAMS special file.
`getmsg(2s)` –	get next message off a Stream.
`getpmsg(2s)` –	get next message off a Stream.
`isastream(2s)` –	test for a STREAMS special file.
`pipe(2s)` –	create a STREAMS pipe.
`poll(2s)` –	wait for an event on a STREAMS file descriptor.
`putmsg(2s)` –	put a message to a STREAMS character device.
`putpmsg(2s)` –	put a band message to a STREAMS character device.
`read(2s)` –	read from a file descriptor.
`readv(2s)` –	read or write a vector.
`write(2s)` –	write to a file descriptor.
`writev(2s)` –	read or write a vector.

The following manual pages are installed in Section 3 of the manual (in the subdirectory `/usr/share/man/man3`):

`streams(3)` –	STREAMS system call library.
`libstreams(3)` –	STREAMS system call library.
`fattach(3)` –	name a STREAMS special file.
`fdetach(3)` –	unname a STREAMS special file.
`isastream(3)` –	test for a STREAMS special file.
`pipe(3)` –	create a STREAMS pipe.
`pstrlog(3)` –	print a STREAMS log buffer.
`s_pipe(3)` –	create a STREAMS pipe.
`strlog(3)` –	print a STREAMS log buffer.
`vstrlog(3)` –	print a STREAMS log buffer.

The following manual pages are installed in Section 4 of the manual (in the subdirectory `/usr/share/man/man4`):

`srvmod(4)` –	STREAMS buffering null module.
`clone(4)` –	the STREAMS clone driver.
`connld(4)` –	STREAMS connection line discipline module.
`conslog(4)` –	STREAMS log device.

[1] The kernel version '`3.0.99-1-unx`' is just an example. For the running kernel, use the output of '`uname -r`'.

echo(4) – echo STREAMS device.
fifo(4s) – STREAMS-based FIFO device.
log(4) – STREAMS log device.
loop(4) – STREAMS loop-around pseudo-device driver.
loop_clone(4) – STREAMS loop-around pseudo-device driver.
mux(4) – STREAMS multiplexing pseudo-device driver.
nsdev(4) – named STREAMS device.
nullmod(4) – STREAMS null module.
nuls(4) – null STREAMS device.
pipe(4) – STREAMS bi-directional pipe device.
pipemod(4) – STREAMS-based pipe module.
s_fifo(4) – STREAMS-based FIFO device.
sad(4) – STREAMS Administrative Driver.
sc(4) – STREAMS Configuration module.
sfx(4) – STREAMS-based FIFO device.
sloop(4) – STREAMS loop-around pseudo-device driver.
spx(4) – STREAMS bi-directional pipe device.
sth(4) – STREAMS Stream head module.
strlog(4) – STREAMS log device.
testmod(4) – STREAMS test module.

The following manual pages are installed in Section 5 of the manual (in the subdirectory /usr/share/man/man5):

autopush(5) – control the autopush module list for a STREAMS device.
specfs(5) – STREAMS special device shadow file system.
strapush(5) – STREAMS autopush structure.
strioctl(5) – STREAMS I/O control data structure.
strsetup.conf(5) – configuration file for STREAMS drivers.

The following manual pages are installed in Section 7 of the manual (in the subdirectory /usr/share/man/man7):

streamio(7) – STREAMS ioctl commands.
I_ANCHOR(7) – STREAMS anchor input-output control.
I_ATMARK(7) – check if a STREAMS message is marked.
I_CANPUT(7) – check if a STREAMS band is writable.
I_CKBAND(7) – check if a STREAMS band is readable.
I_EGETSIG(7) – get enhanced STREAMS SIGPOLL events.
I_ESETSIG(7) – set enhanced STREAMS SIGPOLL events.
I_FATTACH(7) – emulate fattach(2) system call.
I_FDETACH(7) – emulate fdetach(2s) system call.
I_FDINSERT(7) – insert a Stream identifier into a STREAMS message and send it downstream.
I_FIND(7) – find a STREAMS module on a Stream.
I_FLUSH(7) – flush messages from a STREAMS special file.
I_FLUSHBAND(7) – flush messages for a band from a STREAMS special file.
I_GERROPT(7) – get error options for a STREAMS file.
I_GETBAND(7) – get band number of a message on a Stream.
I_GETCLTIME(7) – get close time for a STREAMS file.
I_GETPMSG(7) – STREAMS getpmsg(2s) system call emulation.
I_GETSIG(7) – get SIGPOLL events.
I_GRDOPT(7) – get STREAMS read options.

I_GWROPT(7) – get STREAMS write options.
I_ISASTREAM(7) – emulate isastream(2s) system call.
I_LINK(7) – link a Stream beneath a STREAMS multiplexing driver.
I_LIST(7) – list STREAMS module names on a Stream.
I_LOOK(7) – look at topmost STREAMS module on a Stream.
I_NREAD(7) – number of unread bytes on a Stream.
I_PEEK(7) – peek at STREAMS message on read queue.
I_PIPE(7) – obtain a STREAMS based pipe.
I_PLINK(7) – persistently link a Stream beneath a STREAMS multiplexing driver.
I_POP(7) – pop a STREAMS module from a Stream.
I_PUNLINK(7) – unlink a STREAMS persistent link.
I_PUSH(7) – push a STREAMS module on a Stream.
I_PUTPMSG(7) – STREAMS putpmsg(2s) system call emulation.
I_RECVFD(7) – receive a file descriptor on a Stream.
I_SENDFD(7) – send a file descriptor on a Stream.
I_SERROPT(7) – set error options for a STREAMS file.
I_SETCLTIME(7) – set close time for a STREAMS file.
I_SETSIG(7) – set SIGPOLL events.
I_SRDOPT(7) – set STREAMS read options.
I_STR(7) – STREAMS intput-output control.
I_SWROPT(7) – set STREAMS write options.
I_UNLINK(7) – unlink a Stream from a STREAMS multiplexing driver.

The following manual pages are installed in Section 8 of the manual (in the subdirectory /usr/share/man/man8):

autopush(8) – control the autopush module list for a STREAMS device.
fattach(8) – name a STREAMS file.
fdetach(8) – unname a STREAMS file.
insf(8) – install special device files.
perftest(8) – STREAMS benchmark performance tests on a pipe.
perftestn(8) – STREAMS benchmark performance tests on a pipe.
scls(8) – produce a list of STREAMS module and driver names.
specfs(8) – System V Init Script for the STREAMS Shadow Special Filesystem.
specfs.sh(8) – System V Init Script for the STREAMS Shadow Special Filesystem.
strace(8) – write STREAMS event trace messages to the standard output.
strclean(8) – clean up the STREAMS error logger.
streams(8) – System V Init Script for the STREAMS subsystem.
streams_mknod(8) – create or remove STREAMS device nodes.
streams.sh(8) – System V Init Script for the STREAMS subsystem.
strerr(8) – receive error log messages from the STREAMS log(4) driver.
strinfo(8) – display information about STREAMS devices.
strload(8) – load the STREAMS subsystem.
strsetup(8) – STREAMS setup command.
strvf(8) – STREAMS verification tool.
test-clone(8) – a test suite executable for the clone(4) STREAMS driver.

`test-connld(8)` – a test suite executable for the `connld(4)` STREAMS module.
`test-echo(8)` – a test suite executable for the `echo(4)` STREAMS driver.
`test-fifo(8)` – a test suite executable for the `fifo(4s)` STREAMS driver.
`test-log(8)` – a test suite executable for the `log(4)` STREAMS driver.
`test-loop(8)` – a test suite executable for the `loop(4)` STREAMS driver.
`test-mux(8)` – a test suite executable for the `mux(4)` STREAMS driver.
`test-nsdev(8)` – a test suite executable for the `nsdev(4)` STREAMS driver.
`test-nuls(8)` – a test suite executable for the `nuls(4)` STREAMS driver.
`test-pipe(8)` – a test suite executable for the `pipe(4)` STREAMS driver.
`test-pipemod(8)` – a test suite executable for the `pipemod(4)` STREAMS module.
`test-sad(8)` – a test suite executable for the `sad(4)` STREAMS driver.
`test-sc(8)` – a test suite executable for the `sc(4)` STREAMS module.
`test-streams(8)` – a test suite executable for STREAMS.

The following manual pages are installed in Section 9 of the manual (in the subdirectory `/usr/share/man/man9`):

`Intro(9)` – introduction to STREAMS kernel functions.
`STREAMS(9)` – introduction to STREAMS kernel functions.
`SPG(9)` – *Linux Fast-STREAMS* Programmers Guide.
`DDI(9)` – Device Driver interface/Driver Kernel Interface.
`LfS(9)` – introduction to STREAMS kernel functions.
`mp-streams(9)` – multi-processor STREAMS executive.
`M_BACKDONE(9)` – STREAMS backwash done direct I/O message.
`M_BACKWASH(9)` – STREAMS backwash direct I/O message.
`M_BREAK(9)` – STREAMS break message.
`M_COPYIN(9)` – STREAMS copyin message.
`M_COPYOUT(9)` – STREAMS copyout message.
`M_CTL(9)` – STREAMS control message.
`M_DATA(9)` – STREAMS data message.
`M_DELAY(9)` – STREAMS delay message.
`M_DONTPLAY(9)` – STREAMS don't play direct I/O message.
`M_ERROR(9)` – STREAMS error message.
`M_EVENT(9)` – STREAMS event message.
`M_FLUSH(9)` – STREAMS flush message.
`M_HANGUP(9)` – STREAMS hangup message.
`M_HPDATA(9)` – STREAMS high priority data message.
`M_IOCACK(9)` – STREAMS IO control acknowledgement message.
`M_IOCDATA(9)` – STREAMS IO control data message.
`M_IOCNAK(9)` – STREAMS IO control negative acknowledgement message.
`M_IOCTL(9)` – STREAMS IO control message.
`M_LETSPLAY(9)` – STREAMS let's plan direct I/O message.
`M_NOTIFY(9)` – STREAMS notify message.
`M_PASSFP(9)` – STREAMS pass file pointer message.
`M_PCCTL(9)` – STREAMS priority control message.
`M_PCEVENT(9)` – STREAMS priority event message.
`M_PCPROTO(9)` – STREAMS priority protocol message.
`M_PCRSE(9)` – STREAMS priority reserved message.

M_PCSETOPTS(9) – STREAMS priority set options message.
M_PCSIG(9) – STREAMS priority signal message.
M_PROTO(9) – STREAMS protocol message.
M_READ(9) – STREAMS read message.
M_RSE(9) – STREAMS reserved message.
M_SETOPTS(9) – STREAMS set options message.
M_SIG(9) – STREAMS signal message.
M_START(9) – STREAMS start message.
M_STARTI(9) – STREAMS start input message.
M_STOP(9) – STREAMS stop message.
M_STOPI(9) – STREAMS stop input message.
M_TRAIL(9) – STREAMS trail message.
M_UNHANGUP(9) – STREAMS unhangup message.
OTHERQ(9) – return other queue of a STREAMS queue pair.
QNORM(9) – STREAMS data block structure.
QPCTL(9) – STREAMS data block structure.
RD(9) – return the read queue of a STREAMS queue pair.
SAMESTR(9) – test for a STREAMS pipe or FIFO.
WR(9) – return the write queue of a STREAMS queue pair.
adjmsg(9) – trim bytes from the front or back of a STREAMS message.
allocb(9) – allocate a STREAMS message and data block.
alloclk(9) – allocate or free a STREAMS link block.
allocq(9) – allocate a STREAMS queue pair.
allocstr(9) – allocate a STREAMS Stream head.
appq(9) – append one STREAMS message after another.
apush_get(9) – get the autopush list associated with a STREAMS driver.
apush_set(9) – set the autopush list associated with a STREAMS driver.
apush_vml(9) – verify a STREAMS module list.
autopush(9) – perform autopush operations on a newly opened Stream.
autopush_add(9) – add an autopush list entry for a given STREAMS device number.
autopush_del(9) – delete an autopush list entry for a given STREAMS device number.
autopush_find(9) – find an autopush list entry for a given STREAMS device number.
autopush_search(9) – find an autopush list entry for a given STREAMS device name and number.
autopush_vml(9) – verify a STREAMS module list.
backq(9) – find the upstream or downstream queue.
bcanget(9) – test for message arrival on a band on a Stream.
bcangetany(9) – check whether messages are in any (non-zero) band.
bcanput(9) – test flow control on a STREAMS message queue.
bcanputany(9) – check if a message can be put to any (non-zero) band on a queue.
bcanputnext(9) – test flow control on the next STREAMS message queue.
bcanputnextany(9) – check if a message can be put to any (non-zero) band on the next queue.
bcid_t(9) – install a buffer callback.
bcmp(9) – compare byte strings.

bcopy(9) – copy byte strings.
bufcall(9) – install a buffer callback.
bufcall_id_t(9) – install a buffer callback.
bzero(9) – zero a byte string.
canenable(9) – test whether a STREAMS message queue can be scheduled.
canget(9) – test for message arrival on a Stream.
canput(9) – test flow control on a STREAMS message queue.
canputnext(9) – test flow control on the next STREAMS message queue.
cdev_count(9) – character device switch table helper functions.
cdev_find(9) – character device switch table helper functions.
cdev_match(9) – character device switch table helper functions.
cdev_minor(9) – character device switch table helper functions.
cdev_str(9) – character device switch table helper functions.
cdevsw(9) – the *SVR 4* character device switch table structure.
cdevsw_list(9) – the *SVR 4* character device switch table structure.
cdevsw_lock(9) – the *SVR 4* character device switch table structure.
cdrv_get(9) – the *SVR 4* character device switch table structure.
cdrv_put(9) – the *SVR 4* character device switch table structure.
cmaj_add(9) – major/minor character device node helper functions.
cmaj_del(9) – major/minor character device node helper functions.
cmaj_get(9) – major/minor character device node helper functions.
cmin_add(9) – major/minor character device node helper functions.
cmin_count(9) – major/minor character device node helper functions.
cmin_del(9) – major/minor character device node helper functions.
cmin_find(9) – major/minor character device node helper functions.
cmin_get(9) – major/minor character device node helper functions.
cmin_ini(9) – major/minor character device node helper functions.
cmin_rel(9) – major/minor character device node helper functions.
cmn_err(9) – print a kernel command error.
copyb(9) – copy a STREAMS message block.
copyin(9) – copy user data in from user space to kernel space.
copymsg(9) – copy a STREAMS message.
copyout(9) – copy user data in from kernel space to user space.
copyreq(9) – STREAMS copy request block structure.
copyresp(9) – STREAMS copy response block structure.
cred_t(9) – credentials structure.
ctlmsg(9) – test a STREAMS message type for control.
datab(9) – STREAMS data block structure.
datamsg(9) – test a STREAMS message type for data.
dblk_t(9) – STREAMS data block structure.
delay(9) – postpone the calling process for a number of clock ticks.
dev_t(9) – STREAMS device type.
devnode(9) – STREAMS character device node structure.
do_fattach(9) – implement the fattach(2) system call.
do_fdetach(9) – implement the fdetach(2s) system call.
do_spipe(9) – implement the pipe(2s) system call.
drv_getparm(9) – driver retrieve kernel parameter.

drv_hztomsec(9) – convert kernel tick time between microseconds or milliseconds.
drv_hztousec(9) – convert kernel tick time between microseconds or milliseconds.
drv_msectohz(9) – convert kernel tick time between microseconds or milliseconds.
drv_priv(9) – check if current process is privileged.
drv_usectohz(9) – convert kernel tick time between microseconds or milliseconds.
drv_usecwait(9) – delay for a number of microseconds.
dupb(9) – duplicate a STREAMS message block.
dupmsg(9) – duplicate a STREAMS message.
enableok(9) – allow a STREAMS message queue to be scheduled.
enableq(9) – schedule a STREAMS message queue service procedure.
esballoc(9) – allocate a STREAMS message and data block with caller supplied data buffer.
esbbcall(9) – install a buffer callback for an extended STREAMS message block.
flushband(9) – flushes a band of STREAMS messages from a queue.
flushq(9) – flush messages from a *STERAMS* message queue.
fmod_add(9) – file module switch table helper functions.
fmod_count(9) – file module switch table helper functions.
fmod_del(9) – file module switch table helper functions.
fmod_find(9) – file module switch table helper functions.
fmod_get(9) – file module switch table helper functions.
fmod_put(9) – file module switch table helper functions.
fmod_str(9) – file module switch table helper functions.
fmodsw(9) – the *SVR 4* STREAMS module switch table.
fmodsw_list(9) – the *SVR 4* STREAMS module switch table.
fmodsw_lock(9) – the *SVR 4* STREAMS module switch table.
freeb(9) – frees a STREAMS message block.
freelk(9) – allocate or free a STREAMS link block.
freemsg(9) – frees a STREAMS message.
freeq(9) – deallocate a STREAMS queue pair.
freestr(9) – deallocate a STREAMS Stream head.
freezestr(9) – freeze the state of a Stream.
frtn_t(9) – allocate a STREAMS message and data block with caller supplied data buffer.
getadmin(9) – get the administrative function pointer for a STREAMS module.
getmajor(9) – get the internal major device number for a device.
getmid(9) – get the STREAMS module id for a name.
getminor(9) – get the extended minor device number for a device.
getq(9) – get a message from a STREAMS message queue.
insq(9) – insert a message into a STREAMS message queue.
iocblk(9) – STREAMS input-output control block structure.
isdatablk(9) – test a STREAMS data block for data type.
isdatamsg(9) – test a STREAMS data block for data type.

kmem_alloc(9) – allocate kernel memory.
kmem_alloc_node(9) – allocate kernel memory.
kmem_free(9) – deallocate kernel memory.
kmem_zalloc(9) – allocate and zero kernel memory.
kmem_zalloc_node(9) – allocate and zero kernel memory.
linkb(9) – link a message block to a STREAMS message.
linkblk(9) – STREAMS link block structure.
linkmsg(9) – link a message block to a STREAMS message.
major_t(9) – get the internal major device number for a device.
makedevice(9) – create a device from major and minor device numbers.
max(9) – determine the maximum of two integers.
mblk_t(9) – STREAMS message block structure.
min(9) – determine the minimum of two integers.
minor_t(9) – get the extended minor device number for a device.
modID_t(9) – get the STREAMS module id for a name.
module_info(9) – STREAMS module information structure.
module_stat(9) – STREAMS module statistics structure.
module_stat_t(9) – STREAMS module statistics structure.
msgb(9) – STREAMS message block structure.
msgdsize(9) – calculate the size of the data in a STREAMS message.
msgpullup(9) – pull up bytes in a STREAMS message.
msgsize(9) – calculate the size of the message blocks in a STREAMS message.
noenable(9) – disable a STREAMS message queue from being scheduled.
pcmsg(9) – test a data block message type for priority control.
pullupmsg(9) – pull up the bytes in a STREAMS message.
put(9s) – invoke the put procedure for a STREAMS driver of module with a STREAMS message.
putbq(9) – put a message back on a STREAMS message queue.
putctl(9) – put a control message on a STREAMS message queue.
putctl1(9) – put a one-byte control message on a STREAMS message queue.
putctl2(9) – put a two-byte control message on a STREAMS message queue.
putnext(9) – put a message on the downstream STREAMS message queue.
putnextctl(9) – put a control message on the next STREAMS message queue.
putnextctl1(9) – put a one-byte control message on the next STREAMS message queue.
putnextctl2(9) – put a two-byte control message on the next STREAMS message queue.
putq(9) – put a message on a STREAMS message queue.
qattach(9) – attach a module onto a STREAMS file.
qbackenable(9) – perform back enabling on a STREAMS queue.
qband(9) – queue band structure.
qband_t(9) – queue band structure.
qclose(9) – close a STREAMS driver or module.

qcountstrm(9) – add all counts on all STREAMS message queues in a Stream.
qdelete(9) – delete a queue pair from a Stream.
qdetach(9) – detach a module from a STREAMS file.
qenable(9) – schedule a STREAMS message queue service procedure.
qfields(9) – set attributes of a STREAMS message queue.
qfields_t(9) – set attributes of a STREAMS message queue.
qi_putp(9) – STREAMS driver or module put procedure.
qi_putp_t(9) – STREAMS driver or module put procedure.
qi_qadmin(9) – STREAMS driver or module admin routine.
qi_qadmin_t(9) – STREAMS driver or module admin routine.
qi_qclose(9) – STREAMS driver or module close routine.
qi_qclose_t(9) – STREAMS driver or module close routine.
qi_qopen(9) – STREAMS driver or module open routine.
qi_qopen_t(9) – STREAMS driver or module open routine.
qi_srvp(9) – STREAMS driver or module service procedure.
qi_srvp_t(9) – STREAMS driver or module service procedure.
qinit(9) – STREAMS queue initialization structure.
qinsert(9) – insert a queue pair beneath another queue pair in a Stream.
qopen(9) – call a STREAMS driver or module open routine.
qprocsoff(9) – disable STREAMS message queue processing for multi-processing.
qprocson(9) – enable a STREAMS message queue for multi-processing.
qready(9) – test if queue procedures are scheduled.
qreply(9) – replies to a message from a STREAMS message queue.
qscan(9) – place a queue on the scan list.
qsize(9) – return the number of messages on a queue.
queue(9) – STREAMS message queue structure.
queue_t(9) – STREAMS message queue structure.
register_clone(9) – register a clone(4) minor.
register_cmajor(9) – register external device major number.
register_ioctl32(9) – register a 32-bit IO control command.
register_strdev(9) – register a STREAMS device.
register_strdrv(9) – register a STREAMS driver.
register_strlog(9) – register a STREAMS logger.
register_strmod(9) – register a STREAMS module.
register_strnod(9) – register a STREAMS minor device node.
rmvb(9) – remove a message block from a STREAMS message.
rmvq(9) – remove a message from a STREAMS message queue.
runqueues(9) – run queue service procedures and other asynchronous STREAMS events.
sd_get(9) – acquire and release a reference to the Stream head.
sd_put(9) – acquire and release a reference to the Stream head.
sdev_add(9) – character device switch table helper functions.
sdev_del(9) – character device switch table helper functions.
sdev_get(9) – character device switch table helper functions.
sdev_ini(9) – character device switch table helper functions.
sdev_put(9) – character device switch table helper functions.

sdev_rel(9) – character device switch table helper functions.
sealloc(9) – STREAMS event allocators.
sefree(9) – STREAMS event allocators.
setq(9) – set sizes and procedures associated with a STREAMS message queue.
setqsched(9) – invoke the STREAMS scheduler.
setsq(9) – set synchronization queues, sizes and procedures associated with a STREAMS message queue.
skballoc(9) – allocate a STREAMS message and data block with a caller supplied socket buffer.
spec_open(9) – STREAMS special device shadow file system.
spec_reparent(9) – STREAMS special device shadow file system.
specfs_mount(9) – STREAMS special device shadow file system.
specfs_umount(9) – STREAMS special device shadow file system.
str_close(9) – Stream head module procedures.
str_open(9) – Stream head module procedures.
streamtab(9) – STREAMS module definition structure.
streamtab_t(9) – STREAMS module definition structure.
strgetpmsg(9) – perform a getpmsg(2s) operation on a Stream head.
strioctl(9) – perform a ioctl(2s) operation on a Stream head.
strlog(9) – pass a message to the STREAMS logger.
strm_f_ops(9) – file operations for Stream heads.
stroptions(9) – STREAMS Stream head options structure.
strpoll(9) – perform a poll(2s) operation on a Stream head.
strputpmsg(9) – perform a putpmsg(2s) operation on a Stream head.
strqget(9) – get attributes of a STREAMS message queue.
strqset(9) – set attributes of a STREAMS message queue.
strread(9) – perform a read(2s) operation on a Stream head.
strrput(9) – Stream head module procedures.
strsendpage(9) – perform a sendfile(2s) operation on a Stream head.
strthread(9) – the *SVR 4* STREAMS scheduler thread structure and array.
strthreads(9) – the *SVR 4* STREAMS scheduler thread structure and array.
strwput(9) – Stream head module procedures.
strwrite(9) – perform a write(2s) operation on a Stream head.
strwsrv(9) – Stream head module procedures.
sysctl_str_nstrpush(9) – introduction to STREAMS kernel functions.
sysctl_str_strctlsz(9) – introduction to STREAMS kernel functions.
sysctl_str_strmsgsz(9) – introduction to STREAMS kernel functions.
testb(9) – test if a STREAMS message can be allocated.
timeout(9) – start a timer.
timeout_id_t(9) – start a timer.
timo_fcn_t(9) – start a timer.
toid_t(9) – start a timer.
unbufcall(9) – remove a STREAMS buffer callback.
unfreezestr(9) – thaw the state of a Stream queue.
unlinkb(9) – unlink a message block from a STREAMS message.
unlinkmsg(9) – unlink a message block from a STREAMS message.

unregister_clone(9) –	unregister a clone(4) minor.
unregister_cmajor(9) –	unregister external device major number.
unregister_ioctl32(9) –	unregister a 32-bit IO control command.
unregister_strdev(9) –	unregister a STREAMS device.
unregister_strdrv(9) –	unregister a STREAMS driver.
unregister_strmod(9) –	unregister a STREAMS module.
unregister_strnod(9) –	unregister a STREAMS minor device node.
untimeout(9) –	stop a timer.
unweldq(9) –	unweld two queues.
vcmn_err(9) –	print a kernel command error.
vstrlog(9) –	pass a message to the STREAMS logger.
vstrlog_t(9) –	register a new STREAMS log device.
weld_arg_t(9) –	weld two (or four) queues together.
weld_fcn_t(9) –	weld two (or four) queues together.
weldq(9) –	weld two (or four) queues together.
xmsgsize(9) –	calculate the size of message blocks in a STREAMS message.

5 Porting

OpenSS7 provides a rich set of STREAMS functions, DDI/DKI functions and utilities based on *SVR 4.2 MP* for the development of STREAMS modules and drivers. Although these functions and capabilities provide all of the utilities necessary for the development of STREAMS modules and drivers, it represents the common set of functions provided by other STREAMS implementations.

Some other STREAMS implementations provide interfaces, utilities and helper functions specific to those implementations. Where STREAMS implementations differ the most is in the manner in which they configure and register STREAMS drivers and modules for interface to the operating system, including registration functions, device numbering, creation of minor device nodes, administration and other mechanisms not specified by the *System V Release 4 Programmer's Guide – STREAMS*.

To assist with porting of STREAMS drivers and modules from other STREAMS implementations and *UNIX* based operating systems to *OpenSS7*, *OpenSS7* provides a separate *STREAMS Compatibility* add-on package, called `strcompat-0.9.2.7`,[1] that provide source level compatibility with a wide range of mainstream STREAMS implementations and significant groups of compatibility and helper functions (such as those from *Solaris* and *Mentat*). These compatibility packages also provide separate demand loadable kernel modules that provide the additional compatibility functionality with *OpenSS7*.

In general, when porting to *OpenSS7* from another STREAMS implementation, the following items will need the most attention:

Header Files
: The STREAMS and operating system specific header files that must be included by kernel modules to implement STREAMS drivers or modules are specific to each STREAMS implementation. Although there are some basic header files to include (`sys/stream.h`, `sys/strconf.h`, `sys/ddi.h`, *sys/cmn_err.h*, *sys/dki.h*, *sys/kmem.h*), the order in which these headers are included and the additional operating system specific headers are implementation specific. See the example drivers and modules for the header files that are necessary for *OpenSS7* STREAMS modules and drivers.

Kernel Module Mechanism
: The mechanism for creating, configuring and loading kernel modules is specific to the operating system implementation. *OpenSS7* uses the normal *Linux* mechanisms for kernel modules also for STREAMS drivers and modules.

Configuration and Registration
: The STREAMS driver or module will need to be converted to use the *OpenSS7* configuration and registration mechanisms. See `register_strdev(9)`, `unregister_strdev(9)`, `register_strmod(9)` and `unregister_strmod(9)` for more specific information on the *OpenSS7* configuration and registration mechanisms.

Non-STREAMS DDI/DKI Facilities
: Any of the non-STREAMS DDI/DKI facilities or operating system specific facilities that are used by the STREAMS driver or module may need to be replaced with the *Linux* equivalent. Examples of such facilities include basic locks, read-write locks, semaphores and mutexes, atomic integers, interrupt suppression, bus access and memory mapping functions.

[1] Previously the *STREAMS Compatibility* modules were part of the *OpenSS7* base package. They were separated to an add-on package for a number of reasons that are described in the *OpenSS7 STREAMS Compatibility – Installation and Reference Manual*.

Binary Modules

When STREAMS drivers or modules are released as binary objects and source code is not available, it is still possible to convert the binary module for use with *OpenSS7*. The facility to convert binary modules for use with *OpenSS7* is not, however, part of the base package and is not part of the *STREAMS Compatibility* package. A separate add-on package, the *Binary Compatibility Modules* package, `strbcm-0.9.2.5` was developed explicitly for this purpose.[2]

5.1 Porting from SVR 4.2 MP

When porting from *SVR 4.2 MP* or a STREAMS implementation based closely on *SVR 4.2 MP*, such as *SUPER-UX*, *UXP/V*, *IRIX* or many of the real-time operating system implementations (e.g. *VxWorks*), it is possible to port directly to *OpenSS7* without using the *STREAMS Compatibility* package. Event when porting from *AIX*, *HP-UX* and *OSF/1* it is possible to avoid using the compatibility package.

Most pseudo-device drivers and modules should not require any special facilities beyond basic locks and porting may be straightforward. Where extensive implementation specific DDI/DKI or operating system functions are required, it is better to use the *STREAMS Compatibility* package and modules closest to the specific implementation being ported from.

5.2 Porting from Solaris

When porting from *Solaris* there are both STREAMS facilities and extensive DDI/DKI facilities that differ greatly from basic *SVR 4.2 MP* STREAMS and DDI/DKI functions. For porting all but the most trivial of STREAMS drivers and modules written specifically for *Solaris*, it is better to use the *STREAMS Compatibility* package and the *Solaris* compatibility module provided by that package.[3]

5.3 Porting from UnixWare

When porting from *UnixWare* there are extensive operating system facilities that differ greatly from basic *Linux* facilities. For the most part these are basic locks, read-write locks, condition variables, sleep locks, atomic integers, bus access and mapping functions. Although *Linux* provides equivalents in most of these categories, the *STREAMS Compatibility* package contains a compatibility module for *UnixWare* that provides source compatibility with most of these functions. It is recommended that all but the most trivial of *UnixWare* drivers and modules use the *STREAMS Compatibility* package when porting.

5.4 Porting from Mentat

When porting a STREAMS driver or module from a *Mentat* implementation (such as *AIX*, *HP-UX*, *OSF/1*, *Mac OT*) that makes heavy use of the *Mentat* '`mi_`' or '`mps_`' helper functions, it is best to use the OpenSS7 implementations of those functions available in the *STREAMS Compatibility* package directly. The *STREAMS Compatibility* package provides a *Mentat Portable STREAMS*

[2] Because binary compatibility is a thorny issue, both from the standpoint of technical merit and licensing issues, the `strbcm` package is only currently available to subscribers and sponsors of the *OpenSS7 Project*.

[3] The compatibility package also provides a version of the `strsun.h` header file and the helper functions provided there.

compatibility module that provides implementations of the *Mentat* functions found in *AIX*, *OSF/1* and *Mac OT*.[4]

[4] Note that *Solaris* also provides versions of the *Mentat* functions which appear in the *Solaris* compatiblity module.

6 Conformance

6.1 Standards Compliance

OpenSS7 was designed and implemented to be compliant with as many standards impinging on STREAMS as possible. There are three areas of standards compliance as follows:

6.1.1 User Interface Compliance

The STREAMS user interface standards are primarily specified by the *IEEE* and *OpenGroup* standards and take the form of the *POSIX 2003* and *Single UNIX Specification* standards simultaneously released by the *OpenGroup* in conjunction with *IEEE*. The latest *POSIX/IEEE/OpenGroup* standard provide an *XSI* extension that includes the STREAMS user interface. For the most part, the *OpenGroup* XSI interface is completely compatible with the user interface described in the *System V Release 4 Programmer's Manual – STREAMS*, and where it does not, Stream head options are provided to select between the default *OpenGroup* XSI behaviour and the traditional *SVR 4* behaviour.

Most of the XSI specifications of the *OpenGroup* describe the behaviour of the Stream head and the behaviour of specific STREAMS drivers or modules (such as pipes, FIFOs and terminals). Also described is the `poll(2s)` behaviour, generation of signals, and `read(2s)` and `write(2s)` behaviour as it applies to STREAMS character special devices.

User interface compliance of the *OpenSS7* is tested with custom validation test suites that ship with the package. See Section 3.5.5 [Conformance Test Programs], page 36 for more information on conformance and validation test suites.

6.1.2 Service Interface Compliance

The *OpenGroup* (now and in previous incarnations) have issued standardized service interface specifications as part of the Common Application Environment (CAE) specifications. These service interface specifications usually concern networking interfaces such as the *Data Link Provider Interface (DLPI)*, the *Network Provider Interface (NPI)*, the *Transport Provider Interface (TPI)*, the *X/Open Transport Interface (XTI)* and the *Sockets* API. Although these standards impinge upon various networking add-on packages for *OpenSS7*, they do not impinge upon the base STREAMS package documented here. See the *Installation and Reference Manual* for the appropriate add-on package.

6.1.3 Kernel Interface Compliance

The STREAMS kernel interfaces, DDI/DKI and other facilities available to the STREAMS driver or module writer has not been subjected to formal standardization. For the most part, the descriptions that are present in the *System V Programmer's Manual – STREAMS* provide the most definitive ipso facto standard for STREAMS implementation. In addition to this, some STREAMS implementations have provided some enhancements or restrictions over the *SVR 4* descriptions. Perhaps the most extensive embellishments have been provided for the *Solaris* implementation of STREAMS.

OpenSS7 has been implemented to provide maximum compatibility over a wide range of STREAMS implementations based on *SVR 4* and provides additional capabilities similar to those specific embellishments found in implementations such as *Solaris* through an add-on *STREAMS Compatibility* package.

The most delicate areas of compatibility across STREAMS implementations regard, not the use of STREAMS or DDI/DKI functions from within the STREAMS environment, but the invocation of

STREAMS functions from outside the STREAMS environment. In particular, use of private locks and synchronization in the face of interrupts and external asynchronous callbacks is where implementations deviate the greatest. *OpenSS7* attempts to address these differences by providing a greater level of assurance and wider range of calling contexts for each of the STREAMS facilities.

Kernel interface compliance of the *OpenSS7* to *SVR 4* specifications is tested with custom validation test suites, test modules and test drivers that ship with the package. See Section 3.5.5 [Conformance Test Programs], page 36 for more information on conformance and validation test suites.

6.2 STREAMS Compatibility

OpenSS7 provides a high degree of compatibility with other STREAMS implementation as listed below. Through the separate add-on *STREAMS Compatibility* package, source level compatibility is also provided.

— *SVR 3.2*

OpenSS7 provides a degree of operational compatibility with *SVR 3.2* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *SVR 3.2*.

— *SVR 4.2 ES/MP*

OpenSS7 provides a high degree of operational compatibility with *SVR 4.2 ES/MP* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *SVR 4.2 ES/MP*.

— *Mentat Portable STREAMS*

OpenSS7 provides a high degree of operational compatibility with *Mentat Portable STREAMS* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *Mentat Portable STREAMS*.

— *AIX 5L Version 5.1*

OpenSS7 provides a high degree of operational compatibility with *AIX 5L Version 5.1* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *AIX 5L Version 5.1*.

— *HP-UX 11.0i v2*

OpenSS7 provides a high degree of operational compatibility with *HP-UX 11.0i v2* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *HP-UX 11.0i v2*.

— *OSF/1 1.2/Digital UNIX/True 64*

OpenSS7 provides a high degree of operational compatibility with *OSF/1 1.2/Digital UNIX* to ease portability and common comprehension.

— *UnixWare 7.1.3 (OpenUnix 8)*

OpenSS7 provides a high degree of operational compatibility with *UnixWare 7.1.3 (OpenUnix 8)* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *UnixWare 7.1.3 (OpenUnix 8)*.

— *Solaris 9/SunOS 5.9*
: *OpenSS7* provides a high degree of operational compatibility with *Solaris 9/SunOS 5.9* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *Solaris 9/SunOS 5.9*.

— *IRIX 6.5.17*
: *OpenSS7* provides a high degree of operational compatibility with *IRIX 6.5.17* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *IRIX 6.5.17*.

— *Mac OS 9 Open Transport*
: *OpenSS7* provides a high degree of operational compatibility with *Mac OS 9 Open Transport* to ease portability and common comprehension. Specific kernel utilities are provided by the *STREAMS Compatibility* package to provide full source level compatibility with *Mac OS 9 Open Transport*.

— *SUPER-UX*
: *OpenSS7* provides a high degree of operational compatibility with *SUPER-UX* to ease portability and common comprehension.

— *UXP/V* *OpenSS7* provides a high degree of operational compatibility with *UXP/V* to ease portability and common comprehension.

For additional details, see Section “About This Manual” in *STREAMS Programmer’s Guide*.

7 Releases

This is the OpenSS7 Release of the OpenSS7 core, tools, drivers and modules that implement the *OpenSS7* SVR 4.2 MP STREAMS utility for Linux.
The following sections provide information on OpenSS7 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.

7.1 Prerequisites

When building from source RPM or DSC, the prerequisites for building must be met. Most RPM or DEB build prerequisites are automatic; however, some prerequisites must still be met manually. When building from tarball, most prerequisites must be met manually. The `configure` script will inform you of most missing prerequisites and the actions that must be performed to meet those prerequisites.
Prerequisites for building OpenSS7 package are as follows:

1. *Linux* distribution, somewhat *Linux Standards Base* compliant, with a 2.4, 2.6 or 3.x 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 7.2 [Compatibility], page 62.
   - A fairly LSB compliant GNU/Linux distribution.[1]
   - Linux 2.4 kernel (2.4.10 - 2.4.27),
   - Linux 2.6 kernel (2.6.3 - 2.6.39), or
   - Linux 3.x kernel (3.0 - 3.14);
   - glibc2 or better.
   - GNU groff (for man pages).[2]
   - GNU texinfo (for info files).
   - GNU bison and flex (for config programs).
   - net-snmp (for SNMP agents).[3]
   - GNU gcj and classpath (for Java modules).
   - swig (for Java, Tcl, Perl and Ruby interfaces).

The following will meet most additional prerequisites for a CentOS/RHEL 5.5 build host:

```
#> yum install bzip2 chkconfig coreutils createrepo doxygen \
        gcc-java ghostscript gjdoc glibc gnupg gnuplot \
        groff gzip ImageMagick kernel-devel latex2html \
        libgcj lsof module-init-tools rpm rpm-build tetex \
        tetex-dvips tetex-latex transfig xz xz-lzma-compat \
        zip
```

The following will meet most additional prerequisites for a CentOS/RHEL 6.0 build host:

```
#> yum install bzip2 chkconfig coreutils createrepo doxygen \
gcc-java ghostscript glibc gnuplot groff gzip ImageMagick \
java-1.6.0-openjdk-devel java-1.6.0-openjdk-javadoc \
kabi-whitelists kernel kernel-devel latex2html libgcj \
```

[1] See Section 7.2.1 [GNU/Linux Distributions], page 62, 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] A wide range of net-snmp releases are supported, from UCD-SNMP 4.2.5 through net-snmp 5.7.2.

```
libgcj-devel lsof module-init-tools net-snmp-devel \
perl-devel rpm rpm-build tcl-devel texlive texlive-dvips \
texlive-latex texlive-utils transfig xz zip
```

The following will meet most additional prerequisites for a SuSE Linux Enterprise 11 build host:

```
#> zypper install aaa_base bzip2 coreutils createrepo doxygen \
        fastjar gcc43-gij gcc-java ghostscript-library gjdoc \
        glibc gnuplot gpg2 groff gzip ImageMagick inst-source-utils \
        kernel-default-devel latex2html lsof module-init-tools \
        rpm texlive texlive-latex transfig zip
```

The following will meet most additional prerequisites for a Debian 6.0 build host:

```
#> apt-get install apt-utils bzip2 coreutils createrepo doxygen \
        dpkg dpkg-dev fastjar gcj-jdk gcj-jre-headless ghostscript \
        gnupg gnuplot-nox gnuplot-x11 groff groff-base gzip \
        imagemagick insserv latex2html libc-bin lsof lzma \
        module-init-tools rpm texlive-binaries texlive-font-utils \
        texlive-latex-base transfig xz-utils zip
```

If you need to rebuild the package from sources with modifications, you will need a larger GNU tool chain as described in See Section 8.2.10 [Downloading from CVS], page 128.

7.2 Compatibility

This section discusses compatibility with major prerequisites.

7.2.1 GNU/Linux Distributions

OpenSS7 is compatible with the following *Linux* distributions:[4]

- 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)
- CentOS Enterprise Linux 5.3 (centos53)
- CentOS Enterprise Linux 5.4 (centos54)
- CentOS Enterprise Linux 5.5 (centos55)
- CentOS Enterprise Linux 5.6 (centos56)
- CentOS Enterprise Linux 5.7 (centos57)
- CentOS Enterprise Linux 6.0 (centos60)
- CentOS Enterprise Linux 6.1 (centos61)
- CentOS Enterprise Linux 6.2 (centos61)
- CentOS Enterprise Linux 6.3 (centos61)
- CentOS Enterprise Linux 6.4 (centos61)
- Debian 3.0r2 Woody (deb3.0) TBD

[4] 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.

- 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)
- Debian 5.0 Lenny (deb5.0)
- Debian 6.0 Squeeze (deb6.0)
- Debian 7.0 Wheezy (deb7.0)
- 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)
- Fedora 10 (FC10)
- Fedora 11 (FC11)
- Fedora 12 (FC12)
- Fedora 13 (FC13)
- Fedora 14 (FC14)
- Fedora 15 (FC15)
- 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)
- Mandriva 2010.2 (MDV2010)
- Mandriva Enterprise Server 5.2 (MES52)
- Oracle Linux Server 5.4 (OLS5)
- Oracle Linux Server 5.5 (OLS5)
- Oracle Linux Server 5.6 (OLS5)
- Oracle Linux Server 5.7 (OLS5)
- Oracle Linux Server 6.0 (OLS6)
- Oracle Linux Server 6.1 (OLS6)

- Oracle Linux Server 6.2 (OLS6)
- Oracle Linux Server 6.3 (OLS6)
- Oracle Linux Server 6.4 (OLS6)
- PUIAS Linux 5.4 (PUIAS5)
- PUIAS Linux 5.5 (PUIAS5)
- PUIAS Linux 5.6 (PUIAS5)
- PUIAS Linux 5.7 (PUIAS5)
- PUIAS Linux 6.0 (PUIAS6)
- PUIAS Linux 6.1 (PUIAS6)
- PUIAS Linux 6.2 (PUIAS6)
- PUIAS Linux 6.3 (PUIAS6)
- PUIAS Linux 6.4 (PUIAS6)
- RedHat Enterprise Linux 3.0 (EL3) TBD
- RedHat Enterprise Linux 4 (EL4) TBD
- RedHat Enterprise Linux 5 (EL5)
- RedHat Enterprise Linux 5.1 (EL5)
- RedHat Enterprise Linux 5.2 (EL5)
- RedHat Enterprise Linux 5.3 (EL5)
- RedHat Enterprise Linux 5.4 (EL5)
- RedHat Enterprise Linux 5.5 (EL5)
- RedHat Enterprise Linux 5.6 (EL5)
- RedHat Enterprise Linux 5.7 (EL5)
- RedHat Enterprise Linux 6 (EL6)
- RedHat Enterprise Linux 6.1 (EL6)
- RedHat Enterprise Linux 6.2 (EL6)
- RedHat Enterprise Linux 6.3 (EL6)
- RedHat Enterprise Linux 6.4 (EL6)
- RedHat Linux 7.2 (RH7)
- RedHat Linux 7.3 (RH7)
- RedHat Linux 8.0 (RH8) TBD
- RedHat Linux 9 (RH9) TBD
- Scientific Linux 5 (SL5)
- Scientific Linux 5.1 (SL5)
- Scientific Linux 5.2 (SL5)
- Scientific Linux 5.3 (SL5)
- Scientific Linux 5.4 (SL5)
- Scientific Linux 5.5 (SL5)
- Scientific Linux 5.6 (SL5)
- Scientific Linux 5.7 (SL5)
- Scientific Linux 6.0 (SL6)

- Scientific Linux 6.1 (SL6)
- Scientific Linux 6.2 (SL6)
- Scientific Linux 6.3 (SL6)
- Scientific Linux 6.4 (SL6)
- 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)
- SuSE 11.1 (SuSE11.1)
- SuSE 11.2 (SuSE11.2)
- SuSE 11.3 (SuSE11.3)
- SuSE 11.4 (SuSE11.4)
- SLES 9 (SLES9) TBD
- SLES 9 SP2 (SLES9) TBD
- SLES 9 SP3 (SLES9) TBD
- SLES 10 (SLES10)
- SLES 10 SP1 (SLES10)
- SLES 10 SP2 (SLES10)
- SLES 11 (SLES11)
- SLES 11 SP1 (SLES11)
- SLES 11 SP2 (SLES11)
- SLES 11 SP3 (SLES11)
- 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)
- Ubuntu 8.04 LTS (ubu8.04)
- Ubuntu 8.10 (ubu8.10)
- Ubuntu 9.04 (ubu9.04)
- Ubuntu 9.10 (ubu9.10)
- Ubuntu 10.04 (ubu10.04)
- Ubuntu 10.04.2 LTS (ubu10.04)
- Ubuntu 10.04.3 LTS (ubu10.04)

- Ubuntu 10.10 (ubu10.10)
- Ubuntu 11.04 (ubu11.04)
- WhiteBox Enterprise Linux 3.0 (WBEL3) TBD
- WhiteBox Enterprise Linux 4 (WBEL4) TBD

When installing from the tarball (see Section 8.5.7 [Installing the Tar Ball], page 175), 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.

7.2.2 Kernel

The *OpenSS7* package compiles as *Linux* kernel modules. It is not necessary to patch the *Linux* kernel to build or use the package.[5] Nor do you have to recompile your kernel to build or use the package. OpenSS7 packages use `autoconf(1)` 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(1)` scripts interrogate the kernel for specific features and variants to better adapt to distribution production kernels that have had patches applied over the official kernel.org sources.

7.2.2.1 Compatible Kernels

The *OpenSS7* package is compatible with 2.4 kernel series after 2.4.10 and has been tested up to and including 2.4.33. It has been tested from 2.6.3 up to and including 2.6.38 (with *RHEL 6.0*, *SLES 11.1* and *Debian 6.0* patch sets). It has been tested from 3.0 up to and including 3.14. Please note that your mileage may vary if you use a kernel more recent than 3.2: it is difficult to anticipate changes that kernel developers will make in the future. Many kernels in the 2.6 and 3.x series now vary widely by release version and if you encounter problems, try a kernel within the supported series.

SMP Kernels

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), 3.0GHz Pentium D (2x), dual quad-core SunFire (8x) servers and dual hex-core Xeon servers (12x).

XEN Kernels

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.

7.2.2.2 Linux Kernel Upgrades

The *OpenSS7* package compiles as *Linux* kernel modules. Previously, kernel modules for each and every installed kernel were required. This is no longer the case. The kernel modules installed by the *OpenSS7* package now support weak updates across a wide range of kernels in the series. This

[5] 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.

means that it is normally only necessary to build and install the *OpenSS7* kernel modules for one kernel in a series. This is true for all supported 2.6 and 3.x kernel distributions (and likely for others as well).

7.2.3 Architectures

The *OpenSS7* 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.

7.3 Release Notes

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

Major changes for release openss7-1.1.7.20141001

This is the seventh release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Source code release on GitHub.

Major changes for release openss7-1.1.7.20131209

This is the seventh release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Build corrections for RHEL 6.4.
- Scripts for country boundary, shoreline and rivers database preparation.
- Database for ANSI T1.101 point code assignments.
- Suppress extransous /proc/1/comm error messages when testing for systemd.
- Handle PUIAS to Springdale rename for PUIAS distribution.

Major changes for release openss7-1.1.7.20131123

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Correction to ca-cert handling in RHEL 6 install scriptlet.
- RHEL 6 build changes.
- Set permissions correctly when applying kernel module patch.
- Do not duplicate certs when including ca-cert.

- Avoid conflicting shelll variable in RPM install scriplet.
- Remove /proc/1/comm checks for systemd in module loader.

Major changes for release openss7-1.1.7.20130209

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Corrections to RPM spec file.

Major changes for release openss7-1.1.7.20130129

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Compensate for RHEL6 lack of c_rehash.
- Do not create sysfs nodes: they do not work properly.
- Use no foreground flag in strace.service file.
- More error traces for strerr logger.
- Remove static device names (the do not work either).
- Changes to support weak kernel modules.
- Clean up weak module support.

Major changes for release openss7-1.1.7.20130125

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Better Archlinux install script support.

Major changes for release openss7-1.1.6.20130125

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Remove use of kill_litter_super().

Major changes for release openss7-1.1.3.20130125

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Updates and corrections to weak kernel module builds.
- Repo support and corrections for Archlinux packages.

Major changes for release openss7-1.1.3.20130123

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Additional support for Archlinux.
- Add udev rules for strace and strerr.
- Add system unit files for strace and strerr.
- Add repo support for Archlinux packages.
- Build Archlinux packages better.

Major changes for release openss7-1.1.3.20130121

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.

Major features since the last release are as follows:

- Build support for SLES 11.
- Handle certificates better.
- Get configuration files in the right places for SLES 11.
- Support in RPM spec file for certificates on SLES 11.
- Create a new OpenSS7 certificate.
- Repository support for SLES 11.
- RPM spec file reworked for new RPM release.
- Better certificate handling for RHEL.
- Additional build support for RHEL 6 with certificates.
- Certificate support in Archlinux PKGBUILD.
- Build support for RHEL 6.

Major changes for release openss7-1.1.3.20130111

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.

Major features since the last release are as follows:

- Build and install support for repository.
- Install SSL certificates.
- Cannot open /dev/log anymore.
- Add support for automake-1.13.
- Support systemd modules-load.d and module patching on Archlinux.
- Full detection and support for systemd installations.
- Changes for sysfs device creation and specfs locking.
- Updates to kernel module descriptions and module aliases.
- Use devname aliases where possible.

Major changes for release openss7-1.1.1.20121229

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.

Major features since the last release are as follows:

- Added XMON MIB implementation and active agent.

- Updated GNU build chain to m4-1.4.16, autoconf-2.69, automake-1.12.6, libtool-2.4.2, gettext-0.18.2, flex-2.5.37, bison-2.7, swig-2.0.9, autobuild-5.3, texinfo-4.13a, xz-5.0.4.
- Added documentation for monitoring.
- Added pcapng conversion utilities.
- Added LLDP MIBs.
- Added pcapng library and manual pages.
- Enhanced MIB support.
- Add monitoring and new card support to OPENSS7-MX-MIB.
- Support for Allo cards.
- Updated GNU build chain to m4-1.4.16, autoconf-2.69, automake-1.12.1, libtool-2.4.2, gettext-0.18.1, flex-2.5.35, bison-2.5, swig-2.0.7, autobuild-5.3, texinfo-4.13a, xz-5.0.3.
- Support for Archlinux.
- Support for Slackware.
- Support for 3.0.4-1-lts kernel from Archlinux.
- Support for OpenSS7 Live.
- Support for Debian Wheezy.
- Added (IP) network discovery analyzer.
- Added mxconfig graphical card manager.

Major changes for release openss7-1.1.1.20120725

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Added new OPENSS7-MX-MON-MIB for monitoring (probe).
- Created new X400P-MX driver with full monitoring capabilities to work under Archlinux.

Major changes for release openss7-1.1.1.20120715

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Started new X400P-MX driver to support monitoring and multiple uses of channels with SS7.
- Updates to DS1-EXT-MIB.
- General update of DS0/DS1 MIBS and active agents.
- Corrected bad bug by inspection in pullupmsg().
- Corrected bug in mi_copyin().
- Added BPF driver for tcpdump operation.

Major changes for release openss7-1.1.1.20120708

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.
Major features since the last release are as follows:

- Converted X400P-SS7 driver for automatic line detection and monitoring.

Major changes for release openss7-1.1.1.20120618

This is the sixth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.

Major features since the last release are as follows:

- Correct statistics structures for SS7 link operation and monitoring.
- Build for Archlinux 3.0.36-1-lts kernel.
- Changes to support 3.0 kernel.
- Updated GNU build chain to m4-1.4.16, autoconf-2.69, automake-1.12.6, libtool-2.4.2, gettext-0.18.2, flex-2.5.37, bison-2.7, swig-2.0.9, autobuild-5.3, texinfo-4.13a, xz-5.0.4.
- Updated GNU build chain to m4-1.4.16, autoconf-2.69, automake 1.12.1, libtool 2.4.2, gettext 0.18.1, flex 2.5.35, bison 2.5, swig 2.0.7, autobuild 5.3, texinfo 4.13a, xz 5.0.3.
- Support for Archlinux.
- Support for Slackware.
- Support for 3.0.4-1-lts kernel from Archlinux.
- Support for OpenSS7 Live.

Major changes for release openss7-1.1.1.20110510

This is the fifth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.

Major features since the last release are as follows:

- Support for `repomd` repository.
- Updated GNU build chain to m4-1.4.16, autoconf-2.69, automake 1.11.1, libtool 2.4, gettext 0.18.1, flex 2.5.35, bison 2.5, swig 2.0.4, autobuild 5.3.
- Support for Ubuntu 10.04.2 LTS.
- Support for Mageia 1 (2.6.38.7-desktop-1.mga).
- Support for Mandriva 2010.2.
- Support for Mandriva Linux Enterprise Server 5.2 (2.6.33.7-server-2mnb).
- Support for Scientific Linux 6.0.
- Support for SuSE Linux Enterprise Server 11 SP1.
- Support for OpernSUSE 11.4.
- Support for Fedora 15.
- Support for Red Hat Enterprise Linux Server 6.0.

Major changes for release openss7-1.1.1.20110111

This is the fourth release of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.

Major features since the last release are as follows:

- Support for RHEL 6.0 Beta 2 (refresh) on a 2.6.32-44.1.el6.x86_64(.debug) kernel. Also added support for compiling with gcc 4.4.
- Support for CentOS 5.5 on a 2.6.18-194.25.1.el5 kernel.
- Updated GNU build chain to m4-1.4.15, autoconf-2.69, automake 1.11.1, libtool 2.4, gettext 0.18.1, flex 2.5.35, bison 2.4.3, swig 2.0.1, autobuild 5.3.

- Support for Debian Lenny on a 2.6.26-2 kernel.
- Support for Debian Squeeze on a 2.6.32-5 kernel.
- Preparatory support for Debian Wheezy.

Major changes for release openss7-1.1.1.20090908

This the third *OpenSS7 Project* respin of the *OpenSS7* package. This is a production grade release. All existing validation test suites run clean on supported distributions, production kernels and architectures.

The *OpenSS7* package has undergone significant changes, so follow closely. In it first incarnation, epoch zero (0), the `openss7` package was an add-on package to a now deprecated version of STREAMS. This changed due to the deprecation of the old versions of STREAMS. In its second incarnation, epoch one (1), the `openss7` package contained Linux Fast-STREAMS and a number of other sub-packages, each capable of building and installing separately. This changed due to the increasing maintenance overhead as additional sub-packages were added. In its latest incarnation, epoch two (2), the *OpenSS7* package is a monolithic package containing Linux Fast-STREAMS and most of the other subpackages in a single build package.

Major feature additions since the last public release are:

- Support for CentOS 5.3 and 2.6.18-128.1.10.el5 kernel.
- Support for SLES 10 SP2.
- SNMP agents are more thorough and complete.
- Java JAIN SS7 OAM, MTP, SCCP, TCAP, INAP, MAP, ISUP, ISDN, H.323, IUA, M3UA, SUA, TUA, SPIRITS components, with JAIN JCC 1.2 and MSCONTROL.
- Interfaces provided for major scripting languages including Java, Tcl, Perl and Ruby using Swig.

Major packaging changes since the last public release, epoch one (1), are:

- The package only compiles with Linux Fast-STREAMS, which is included.
- Only OpenSS7 Project developed modules, drivers, libraries and utilities have been included.
- Forks of other test utilities such as Iperf and Netperf have been removed.
- Sub-package source and installation sub-directories have all been collapsed into the one `openss7` sub-directory.
- A single tarball, SRPM and DSC generates a simple set of 3 binary RPMs or DEBs.
- The entire package is now released under the Affero GNU Public License Version 3; however, commercial licensing is also available for the entire package.

Major upgrades from the last public release are:

- Updated build system to `swig-1.3.39`.
- Updated build system to `automake-1.11` with full silent build.
- Excluded deprecated sub-packages as follows:
  - `sctp-0.2.27`
  - `iperf-2.0.8`
  - `netperf-2.3.7`
  - `osr61-0.9.2.3`
- Documentation license for all documentation upgraded to GFDL 1.3 with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts, for full Debian policy compatibility.

- Updated build system tool chain to `m4-1.4.13`, `automake-1.10.2`, `bison-2.4.1`, `libtool-2.2.6a`, `texinfo-4.13a`.
- Addition of the following sub-packages:
  - `stratm-0.9.2.1`

  This subpackage was then instantly absorbed into the new *OpenSS7* package.
- Significant feature updates were made to:
  - `strchan-0.9.2.4`
  - `strsock-0.9.2.4`
  - `strnsl-0.9.2.4`
  - `sigtran-0.9.2.4`
- Significant rework of the `strchan` sub-package into the `strss7`, `strisdn`, `strx25` and `striso` packages. One single driver now meets the needs of all sub-packages.
- Completion of the `strsock` package, moves from pre-alpha to beta.
- The `strnsl` subpackage moves from alpha to production.
- Significant completion of the `sigtran` package: all components moved back to production.

Major maintenance items since the last public release are:

- Fixed hash races in open and close. This was service affecting under heavy open/close loads. (Bug# openss7 006.)
- Fixed reference counting imbalance when entering and leaving synchronization queues. (Bug# openss7 005.)
- Fixed reference counting imbalance when syncrhonization queues are used on module. (Bug# openss7 004.)
- Fixed timer reporting in the pstrlog() call resulting in near-epoch invalid timestamps. This was an annoyance. (Bug# openss7 003.)
- Fixed assurances in untimeout and unbufcall utilities. This could be service affecting on many-way machines. (Bug# openss7 002.)
- Timer cancellation directly from ICS races removed. This could be service affecting on many-way machines. (Bug# openss7 002.)
- Corrected removal of SLAP_NO_REAP flag from recent kernels causing timeout problems on many-way machines under memory pressure. (Bug# openss7 001.)

For complete detail on maintenance items, see `BUGS` in the release.

This is a public, stable, production grade release of the package: it deprecates previous releases. Please upgrade to the current release before reporting bugs.

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(8)` and `dpkg(1)` based distributions, and can be used on production kernels without patching or recompiling the kernel.

This package is publicly released strictly under the *GNU Affero General Public License Version 3*; however, commercial licensing is available. The release is available as an `autoconf(1)` tarball, SRPM, DSC, and a set of binary RPMs and DEBs for popular distributions. See the package page for the `autoconf(1)` tarballs, SRPMs, DSCs and binary RPMs and DEBs. See the repository page for information concerning network installation and update sources.

See `http://www.openss7.org/codefiles/openss7-1.1.7.20141001/ChangeLog` and `http://www.openss7.org/codefiles/openss7-1.1.7.20141001/NEWS` in the release for more information. Also, see the `openss7.pdf` manual in the release (also in html).

For the news release, see `http://www.openss7.org/rel2009XXXX_L.html`.

Major changes for release openss7-1.1.1.20090901

This release was only released internally.

Major changes for release openss7-1.1.1.20090622

This release was only released internally.

Major changes for release openss7-0.9.2.G

This is the sixth public release of the OpenSS7. See `README` in the release for a sub-package listing. Most of the sub-packages in the release are production grade for *Linux Fast-STREAMS*. All existing validation test suites run clean on supported distributions and architectures.

The *OpenSS7* is not released as often as the sub-packages. As sub-packages are released more often, to rebuild the master package with a new sub-package release, simply replace the directory to which the sub-package belongs with the unpacked sub-package release and then rebuild the master package. This release provides support for recent distributions and tool chains.

Major features since the last public release are as follows:

- License upgrade to *AGPL Version 3*.
- Modifications to build under *Fedora* '`2.6.22.5-49`' kernel. These changes also support '`2.6.22.9-91.fc7`' kernel. Modifications to build under *Fedora* '`2.6.25-45.fc9`' and '`2.6.26.5-45.fc9`' kernels. Documented `lib32gcc1` problem on Ubuntu. Noted problem running under XEN kernels. XEN kernels are not yet supported. Added `MODULE_VERSION` to all modules and drivers.
- Ability to strap out major documentation build and installation primarily for embedded targets. Improvements to common build process for embedded and cross-compile targets. Cross-compile fixes (strap out `AC_FUNC_REALLOC` macro when cross-compiling). Conversion of RPM spec files to common approach for major subpackages.

  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.
- Higher performance and updated performance papers.
- Updated tool chain to `m4-1.4.12`, `autoconf-2.63` and `texinfo-4.13`. Support for `flex 2.5.33` in maintainer mode.
- Updated references database for manual pages and roff documents.
- Added the following major sub-packages to the master build (and release):
  - `strx25-0.9.2.1`
- All of the major subpackages that are at production release have maintenance upgrades for new production kernels, distributions and tool chains. Many of the major subpackages have greatly expanded documentation and provide additional modules and drivers not previously available in public releases. Significant feature updates were made to:
  - `strxns-0.9.2.7`
  - `strinet-0.9.2.7`
  - `strchan-0.9.2.4`
  - `strx25-0.9.2.1`
  - `striso-0.9.2.4`
  - `strss7-0.9a.8`
  - `sigtran-0.9.2.4`

  Please see the individual `NEWS` files in each of the subpackages for more information.

This is a public stable production grade release of the package: it deprecates previous releases. Please upgrade to the current release before reporting bugs.

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 openss7 package page.

See `http://www.openss7.org/codefiles/openss7-1.1.7.20141001/ChangeLog` and `http://www.openss7.org/codefiles/openss7-1.1.7.20141001/NEWS` in the release for more information. Also, see the `openss7.pdf` manual in the release (also in html `http://www.openss7.org/openss7_manual.html`).

For the news release, see `http://www.openss7.org/rel20081029_L.html`.

Major changes for release openss7-0.9.2.F

This is the fifth public release of the OpenSS7. See `README` in the release for a sub-package listing. Most of the sub-packages in the release are production grade for *Linux Fast-STREAMS*. All existing validation test suites run clean on supported distributions and architectures.

The *OpenSS7* is not released as often as the sub-packages. As sub-packages are released more often, to rebuild the master package with a new sub-package release, simply replace the directory to which the sub-package belongs with the unpacked sub-package release and then rebuild the master package. This release provides support for 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 and 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 openss7-0.9.2.E

This is the fourth public release of the OpenSS7. See `README` in the release for a sub-package listing. Most of the sub-packages in the release are production grade for *Linux Fast-STREAMS*. All existing validation test suites run clean on supported distributions and architectures.

It is unlikely that the *OpenSS7* will be released as frequently as before. Sub-packages will be released more often. To rebuild the master package with a new sub-package release, simply replace the directory to which the sub-package belongs with the unpacked sub-package release and then rebuild the master package. This release provides support for recent distributions and tool chains.

Major features since the last public release are as follows:

- Addition of the `osr61` sub-package that contains Dialogic® Open System Release 6.1 version 239 GPL drivers.
- A few minor corrections to the common build process.

- Support for autoconf 2.61, automake 1.10 and gettext 0.16.
- Support for Ubuntu 6.10 distribution and bug fixes for i386 kernels.
- The package now looks for subpackages with a version number as unpacked by separate tarball.

Major changes for release openss7-0.9.2.D

This is the fourth public release of the OpenSS7. The sub-packages have been reorganized for this release. See `README` in the release for a sub-package listing. Aside from sub-package reorganization, the major difference from previous release is that this release no longer contains *LiS*. Too many of the sub-packages will not even build against *LiS* because of its Stream head deficiencies.

Most of the sub-packages in the release are production grade for *Linux Fast-STREAMS*. All existing validation test suites run clean on supported distributions and architectures. The packages build better Debian/Ubuntu .deb files.

It is unlikely that the *OpenSS7* will be released as frequently as before. Sub-packages will be released more often. To rebuild the master package with a new sub-package release, simply replace the directory to which the sub-package belongs with the unpacked sub-package release and then rebuild the master package.

The release provides the following enhancements and fixes:

- Added the following sub-packages to the master build (and release):
  - `strnsl-0.9.2.1`
  - `strbcm-0.9.2.1`
  - `striso-0.9.2.1`
  - `strsock-0.9.2.1`
  - `strtty-0.9.2.1`
  - `strutil-0.9.2.1`
- Automated release file generation making for vastly improved and timely text documentation present in the release directory.
- This release candidate includes the changes made to the strsctp drivers at the 2006 SCTP Interop at the University of British Columbia. This version was interoperability tested with all implementations present.
- Better support for Ubuntu and recent `gcc` compilers, including debian script corrections.
- Support for most recent 2.6.18 kernels (including Fedora Core 5 with inode diet patchset).
- Now builds 32-bit compatibility libraries and tests them against 64-bit kernel modules and drivers. The '`make installcheck`' target will now automatically test both 64-bit native and 32-bit compatibility versions, one after the other, on 64-bit platforms.
- Added versions to all library symbols.
- Many documentation updates for all OpenSS7 packages.
- Dropped support for *LiS*.
- Start assigning majors at major device number 231 instead of major device number 230. Assign major device number 230 explicitly to the clone device. Package will now support extended ranges of minor devices on 2.6 kernels under Linux Fast-STREAMS only. `streams` now supports expanded addressable minor device numbers, permitting 2^16 addressable minor devices per major device number on 2.6 kernels: LiS cannot support this change.
- Better detection of SuSE distributions, release numbers and SLES distributions: support for additional SuSE distributions on ix86 as well as x86_64. Added distribution support includes SLES 9, SLES 9 SP2, SLES 9 SP3, SLES 10, SuSE 10.1.

- Improvide compiler flag generation and optimizations for recent `gcc` compilers and some idiosyncratic beahviour for some distributions (primarily SuSE).
- Optimized compilation is now available also for user level programs in addition to kernel programs. Added new '`--with-optimize`' option to `configure` to accomplish this.
- Added '`--disable-devel`' `configure` option to suppress building and installing development environment. This feature is for embedded or pure runtime targets that do not need the development environment (static libraries, manual pages, documentation).
- Added `send-pr` script for automatic problem report generation.
- Each package will not build `doxygen(1)` html documentation with the '`make doxy`' target. See '`make help`' or `README-make` in the distribution for more information.

Major changes for release openss7-0.9.2.D.rc3

Third release candidate.

- The package will now build doxygen(1) html documentation with the 'doxy' make target. See 'make help' or README-make in the distribution for more information.
- Now builds 32-bit compatibility libraries and tests them against 64-bit kernel modules and drivers. The '`make installcheck`' target will now automatically test both 64-bit native and 32-bit compatibility versions, one after the other, on 64-bit platforms.
- Added complete documentation and *Installation and Reference Manual* for the *OpenSS7* (this manual).
- Added the following sub-packages to the master build (and release):
  - `strbcm-0.9.2.1`
  - `striso-0.9.2.1`
  - `strsock-0.9.2.1`
  - `strtty-0.9.2.1`
  - `strutil-0.9.2.1`
- Automated release file generation making for vastly improved and timely text documentation present in the release directory.
- Dropped support for *LiS*.
- Sub-packages will now support extended ranges of minor devices on 2.6 kernels under *Linux Fast-STREAMS* only.
- This release candidate provides support for additional SuSE distributions on ix86 as well as x86_64. Added distribution support includes SLES 9, SLES 9 SP2, SLES 9 SP3, SLES 10, SuSE 10.1.
- This release candidate includes the changes made to the strsctp drivers at the 2006 SCTP Interop at the University of British Columbia. This version was interoperability tested with all implementations present.

This was an subscriber release.

Major changes for release openss7-0.9.2.D.rc2

Second release candidate.

This release candidate also contains the results of performance testing of the new second generation UDP driver (implemented completely in STREAMS instead of using an internal socket).

This release candidate also contains support for SuSE 10.1.

This was an subscriber release.

Major changes for release openss7-0.9.2.Drc1

First release candidate.

- Release candidate for Mark Fugate.
- Added –enable-devel configure option for embedded targets.
- Added send-pr script for automatic problem report generation.

This was an subscriber release.

Major changes for release openss7-0.9.2.C

Distribution check for entire master package. Trying to get master package into form where it can be released as a complete package.
This was a public release.

Major changes for release openss7-0.9.2.B

Minor changes for wider release, better master packaging and bug fixes.
This was a public release.

Major changes for release openss7-0.9.2.A

With this release version numbers were changed to reflect an upstream version only to be consistent with other OpenSS7 package releases. All RPM release numbers will be -1$(PACKAGE_RPMEXTRA) and all Debian release numbers will be '_0'. If you wish to apply patches and release the package, please bump up the release number and apply a suitable release suffix for your organization. We leave Debian release number _1 reserved for your use, so you can still bundle the source in the .dsc file.
Major changes for this release include build against Linux 2.6 kernels and popular distributions based on the 2.6 kernel as well as wider distribution support.
This was a public release.

Initial release openss7-0.9.2-1

Initial autoconf/RPM release of the OpenSS7 master package.
This master package contains all other OpenSS7 releases.

7.4 Maturity

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

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

7.4.1 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.

7.4.2 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.

7.4.3 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.

7.4.4 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.

7.4.5 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.

7.4.6 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.

7.5 Bugs

7.5.1 Defect Notices

OpenSS7 could contain unknown defects. This is a *beta* release. Some defects could be harmful. Validation testing has been performed by the *OpenSS7 Project* on this software for only a restricted set of systems. The software might fail to configure or compile on other systems. The *OpenSS7 Project* recommends that you **do not use this software for purposes other than validation testing and evaluation, and then only with care**. Use at your own risk. Remember that there is **NO WARRANTY**.[6]

This software is *beta* software. As such, it might crash your kernel. Installation of the software might 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 might crash your kernel, the resulting unstable system could possibly destroy computer hardware or peripherals making them unusable. You might void the warranty on any system on which you run this software. YOU HAVE BEEN WARNED.

7.5.2 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* had no known bugs at the time of release.

7.5.3 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

Some portions of this package are not completely implemented yet, so the bugs for those portions are still being designed and will not be available until a later date. Most of the package is complete and for the most part bugs are currently available.

[6] See sections **Disclaimer of Warranty** and **Limitation of Liability** under [GNU Affero General Public License], page 201.

`openss7 009. 2010-12-14T07:29:58+0000`

Found a bug in msgpullup(9) that caused it to always fail. Strange enough, msgpullup(9) only attempted after pullupmsg(9) failed in all OpenSS7 drivers and modules.

fixed in `openss7-1.1.1.20110111`

`openss7 008. 2010-08-09T17:41:46+0000`

The `t_alloc(3)` function was incorrectly returning a *TLOOK* error when called when an outstanding event was pending. Thanks to John Hodgkinson at Boldon James for reporting this bug. A similar bug was found by inspection for `t_bind(3)`, `t_getinfo(3)`, `t_getprotaddr(3)`.

fixed in `openss7-1.1.1.20110111`

`openss7 007. 2010-06-09T12:13:41+0000`

The `t_rcv(3)` function was not reseting data when the amount of data to be read was exactly the amount of data available. Thanks to John Hodgkinson at Boldon James for reporting this bug.

fixed in `openss7-1.1.1.20111011`

`openss7 006. 2009-08-31T04:04:52+0000`

A number of functions in `strlookup.c` were moving the found item in the hash collision list to the head of the list; however, a number of these functions were being called with a read lock instead of a write lock on the corresponding hash table. This caused crashes on kernels checking for list corruption on SMP machines under heavy open/close loads. The movement of the found entry to the head of the collision list has dubious performance advantages (as the hash entries are cached). Therefore, the practice has been removed (actually suppressed). Thanks to Tony Abo at HiTech for the report leading to this bug.

fixed in `openss7-1.1.1.20090908`

`openss7 005. 2009-06-13T03:47:32+0000`

There was an `qput(9)`/`qget(9)` reference counting imbalance in the `enter_syncq_writer(9)`, `enter_inner_syncq_func(9)` and `enter_inner_syncq_putp(9)` functions. The most symptomatic of these was the `enter_inner_syncq_putp(9)` function that affected regular `put(9s)` and `putnext(9)` functions on synchronized modules and drivers having an inner perimeter. The reference imbalance resulted in leaking the module queue pair, stream head and stream head queue pair on driver close.

fixed in `openss7-1.1.1.20090908`

`openss7 004. 2009-06-12T08:16:08+0000`

`qattach(9)` was taking a reference on a synchronization queue without the reference being released in `qdelete(9)`. `qdelete(9)` was fixed to release the reference taken by `setsq(9)` in `qattach(9)`. Thanks to Tony Abo at HiTech for identifying and reporting this bug.

fixed in `openss7-1.1.1.20090908`

`openss7 003. 2009-04-17T11:56:06+0000`

The strace, strerr daemons and the pstrlog call in the streams library were not initializing ltime before the call to `ctime_r(3)` resulting in an epoch timestamp (Jan 1, 1970). Thanks to Pierre Crepieux for identifying and reporting this bug.

fixed in `openss7-1.1.1.20090908`

`openss7 002. 2009-03-26T07:56:40+0000`

Untimeout and unbufcall did not wait until the callback returned before returning when there was a collision between a cancellation of an event and the callback for the event. Linux Fast-STREAMS now makes this assurance, provided that the cancellation is not being invoked from within the same thread as the callback (i.e. untimeout called from an ISR interrupting the callback, or, say, from the callback itself) in which case it returns immediately.

Also, additional timer problems were encountered. Cancelling timers from within an ISR did not have sufficient list protection (irq suppression) potentially resulting in list corruption or queue reference counting problem.

fixed in `openss7-1.1.1.20090908`

`openss7 001. 2008-12-16T08:17:47+0000`

Somewhere about Linux kernel '`2.6.17`', and during the `openss7-0.9.2.D.rc2` master package release, it was discovered that SLABs no longer supported the SLAB_NO_REAP flag. Unfortunately, the `seinfo_ctor(9)` function was assuming that the SLAB_NO_REAP flag was being recognized. This means that over the span of some several days on system heavily using timers that a slab corruption would eventually occur resulting in a kernel crash, particularly on x86_64 kernels. The '`seinfo`' slab functions have been rewritten to not expect the SLAB_NO_REAP feature. There was also a minor possibility of a strevent structure identifier overlap after an extremely long period of intensive operation, and that has been fixed as well. Thanks to Angel Diaz for reporting this bug.

fixed in `openss7-1.1.1.20090908`

`strinet 004. 2008-10-24T08:04:16+0000`

UDP and RAWIP drivers cannot receive zero-length messages in accordances with UNIX '98 and XNS 5.2. Use the UDP2 and RAWIP2 drivers instead if this is important to you as they exhibit full UNIX '98 XNS 5.2 conformance.

fixed in `strinet-0.9.2.7`

`strcompat 008. 2008-10-19T19:57:41+0000`

`mi_open_link(9)` was not walking device lists correctly.

fixed in `strcompat-0.9.2.7`.

`strcompat 007. 2008-10-19T10:39:26+0000`

`mi_open_link(9)` was not returning the assigned device number in devp when sflag was CLONEOPEN.

fixed in `strcompat-0.9.2.7`.

`streams 025. 2008-10-17T05:57:29+0000`

'`putnext(q, mp)`' was checking whether procedures had been turned off on queue '`q`'. This was not correct as it is only the '`q->q_next`' put procedure that would be executed. It should only check procedures on '`q->q_next`'.

fixed in `streams-0.9.2.4`

`streams 024. 2008-10-11T19:36:41+0000`

A list delete corruption bug in the STREAMS driver and module lookup functions (e.g. `__cdrv_lookup`) was discovered by the list debugging in the *FC9* kernel.

fixed in `streams-0.9.2.4`

`streams 023. 2008-10-11T19:36:23+0000`

Not really a bug, but newer (2.6.25) kernels no longer permit registration of binary identifiers for sysctls (i.e. *ctl_name*). The proc filesystem entries (i.e. *procname*) are still permitted and *ctl_name* should be set to zero for these kernels. Added a check for the existence of symbol `sysctl_check_table(9)` to identify when binary registration is forbidden. Another related problem is that when binary registration of system controls is not possible, `sysctl(2)` becomes worthless. Unfortunately, the STREAMS MIB agent was written to use `sysctl(2)` and needs to be rewritten to use the `/proc/sys` filesystem instead ala `sysctl(8)`.

fixed in `streams-0.9.2.4`

`streams 022. 2008-10-07T18:40:25+0000`

When overriding 32-bit compatability on input-output controls conflicting from the CDROM block device with STREAMS input-output controls, the override was not properly passing CDROM input-output controls through due to a missing break statement in the override loop. This bug affected pre-2.6.11 kernels, likely manifesting itself in a non-function CDROM device while STREAMS was loaded. Bug reported and one-line fix provided by Sylvain Chouleur for *DGAC*.

fixed in `streams-0.9.2.4`

`strxns 003. 2008-09-03T06:10:28+0000`

Over-restricted settable range of tos in `np_ip` driver. Thanks to Christophe Nolibos for reporting this bug.

fixed in `strxns-0.9.2.7`

`strnsl 004. 2008-08-20T06:02:09+0000`

To permit the xnsl library to be used on systems with the broken libtirpc package, we now place the file in /etc/netconfig.xnsl instead of /etc/netconfig where the broken libtirpc package for GNU/Linux expects a broken /etc/netconfig configuration file. The library has been adjusted to first look for /etc/netconfig.xnsl before looking for /etc/netconfig.

fixed in `strnsl-0.9.2.4`.

`strss7 006. 2008-08-17T03:42:23+0000`

The xnet library was discovered to contain a thread-safety bug caused by newer behaviour of `pthread_once(3)` causing the library to core dump when used on recent implementations of pthreads (nptl).

The xnet library is no longer normally compiled as part of the `strss7` package.

fixed in `strss7-0.9a.8`.

`strsock 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(3)` causing the libraries to core dump when used on recent implementations of pthreads (nptl).

fixed in `strsock-0.9.2.4`.

`strnsl 003. 2008-08-17T03:42:23+0000`

The xnsl library was discovered to contain a thread-safety bug caused by newer behaviour of `pthread_once(9)` causing the library to core dump when used on recent implementations of pthreads (nptl).

fixed in `strnsl-0.9.2.4`.

`strxnet 001. 2008-08-17T03:42:23+0000`
: The xnet and xnsl libraries were discovered to contain a thread-safety bug caused by newer behaviour of `pthread_once(3)` causing the libraries to core dump when used on recent implementations of pthreads (nptl).

  fixed in `strxnet-0.9.2.12`.

`strinet 003. 2008-08-02T02:58:49+0000`
: Some test cases are failing for the TCP driver when SELinux is set for Enforcing. Either the STREAMS kernel threads kstream/0, etc. need to be permitted or SELinux must be set to Permissive or Disabled. Conditions were added to the testsuite to expect failures on 30 specific tests when SELinux is set to Enforcing.

  addressed in `strinet-0.9.2.7`

`streams 021. 2008-08-01T22:32:08+0000`
: When flushing queues the backenable bits were not being initialized to zero in `__flushq(9)`, resulting in back-enabling of bands (or the normal queue) was being performed depending on the uninitialized values in the backenable bit array. This only affected `I_SETSIG` signals for `SWRNORM` and `SWBAND`, and the only when flushing queues. Fix properly initializes the backenable array.

  fixed in `streams-0.9.2.4`

`streams 020. 2008-07-31T04:59:41+0000`
: Not really a bug (for STREAMS), but when the `streams.ko` kernel module is loaded, the `crash(8)` debugger will not debug a running kernel because it finds the `runqueues(9)` exported function in the `streams.ko` module instead of the the static one from the kernel. This has been temporarily renamed by macro to `srunqueues(9)` (notice the leading '`s`') until `crash(8)` learns to do the right thing and check that the symbol it looks up comes from the kernel instead of a kernel module.

  workaround in `streams-0.9.2.4`

`streams 019. 2008-07-25T22:41:47+0000`
: When `M_READ` was being issued by the Stream Head downstream an `srlock(0)` imbalance in `strsendmread(9)` was causing soft-lockups on close for recent read-write lock implementations on *CentOS 5.2* for '`x86_64`'.

  fixed in `streams-0.9.2.4`

`streams 018. 2008-07-25T01:15:26+0000`
: Previous fix didn't work too good: returning `[EAGAIN]` when hung-up on `getmsg(2s)`, `getpmsg(2s)`, `read(2s)`, `readv(2s)` instead of 0 and terminal end of file. This caused a regression on four or five other test cases.

  fixed in `streams-0.9.2.4`

`strcompat 006. 2008-07-11T13:52:04+0000`
: lis_alloc_sem() was not setting the supplied count against the created semaphore, but was alway setting the semaphore to 1 (unlocked).

  fixed in `strcompat-0.9.2.7`.

`strcompat 005. 2008-05-26T14:06:22+0000`
: lis_register_strdev() was failing whenever nminor was greater than zero or not an even multiple of 256. This was due to an error in the logic checking for multiple majors. (Thanks to Omer Tunali for reporting this bug.)

  fixed in `strcompat-0.9.2.7`.

`streams 017. 2008-04-10T15:17:30+0000`

When `M_DATA` is sent upstream followed by `M_HANGUP`, `read(2s)` is returning zero (0) and not permitting the data associated with the `M_DATA` to be read. This is a bug per documentation. `read(2s)` should operate as normal following a hangup until all data is read and then return zero (0).

The difficulty is that when waking up from a read sleep or when entering read the hangup condition was generating an internal `[ESTRPIPE]` error. This was altered so that `[ESTRPIPE]` is only returned during the hangup condition after the read queue has been tested and the caller is about to sleep on read.

Test cases 3.2.1, 3.5.1 and 3.6.1 in the test-streams test suite executable were altered to validate the fix for this case and curtail regressions.

fixed in `streams-0.9.2.4`

`streams 016. 2007-11-14T17:23:57+0000`

Read is blocking when data has been read, `O_NONBLOCK` and `O_NDELAY` unset, `RFILL` unset, in non-SVR4 mode. This violates POSIX specifications.

Test case 3.1.11.4 in the test-streams test suite executable was generated to validate the fix for this case and to curtail regressions.

fixed in `streams-0.9.2.4`

`streams 015. 2007-11-14T17:19:01+0000`

Dynamic allocation of major device numbers is not working on recent 2.6 kernels. Someone slipped some code in the kernel to have `register_chrdev(9)` allocate from major 255 down (again). Changed code to allocate modid according to our own rules and then request the same for a major device number. This also ensures that module ID and major are the same.

fixed in `streams-0.9.2.4`

`strchan 001. 2007-10-15T16:17:08+0000`

Removing definition of `freezestr(9)` and `unfreezestr(9)` in ch_pmod.c and mx_pmod.c was causing loss of version information on old 2.4 kernels. Workaround is to use the _SUN versions as is until it can be fixed in streams or strcompat.

fixed in `strchan-0.9.2.4`

`strsctp 007. 2007-07-21T21:06:48-0600`

It was discovered that many network devices were claiming to have hardware checksum capability by setting the *NETIF_HW_CSUM* flag, but did not support SCTP checksum offload.

workaround in `strsctp-0.9.2.8`

The workaround is to ignore the flag for now. Either Linux network drivers need to start supporting SCTP checksum when they set the *NETIF_HW_CSUM* flag, or there needs to be some flag provided for SCTP. The workaround is to not believe that the driver does SCTP checksum offload when *NETIF_HW_CSUM* is set.

`strsctp 006. 2007-07-21T21:05:10-0600`

It was detected that the *XTI_SNDBUF* and *XTI_RCVBUF* options were not being set correctly (in fact, not being set at all).

fixed in `strsctp-0.9.2.8`

XTI_SNDBUF now limits the maximum available send window. *XTI_RCVBUF* now alters the advertized receive window.

`openss7 002. 2007-07-21T17:26:01-0600`
: It was reported that validation test suites for XEN kernels are failing. XEN kernels are, therefore, not supported. (Thanks to Bryan Shupe at Flying J for reporting this bug.)

  noted in `openss7-0.9.2.G`

`strcompat 004. 2007-07-21T17:26:01-0600`
: It was reported that, even with the fix below, validation test suites for XEN kernels are failing. XEN kernels are, therefore, not supported. (Thanks to Bryan Shupe at Flying J for reporting this bug.)

  noted in `strcompat-0.9a.7.rc1`.

`strcompat 003. 2007-07-21T17:22:10-0600`
: It was reported that paddr_t is already defined in recent XEN kernels, causing compile to fail for these kernels. (Thanks to Bryan Shupe at Flying J for reporting this bug.)

  fixed in `strcompat-0.9a.7.rc1`.

  A check was added to the configure script to check for the existence of paddr_t.

`strss7 005. 2007-07-21T17:15:02-0600`
: It was discovered that recent kernel on RHAS4 are defining irq_handler_t but have the old 3 argument function template for irq handlers. The detection logic assumed that if irq_hander_t existed, that the newer 2 argument function template for irq handlers were in effect. This caused builds to fail on these RHAS4 kernels.

  fixed in `strss7-0.9a.8.rc1`.

  A check was added to the configure script to test whether the irq_handler_t has the newer 2 argument template.

`strcompat 002. 2007-07-21T17:15:02-0600`
: It was discovered that recent kernel on RHAS4 are defining irq_handler_t but have the old 3 argument function template for irq handlers. The detection logic assumed that if irq_hander_t existed, that the newer 2 argument function template for irq handlers were in effect. This caused builds to fail on these RHAS4 kernels.

  fixed in `strcompat-0.9a.7.rc1`.

  A check was added to the configure script to test whether the irq_handler_t has the newer 2 argument template.

`openss7 001. 2007-06-27T08:53:51+0000`
: A report was made by Chris from Sandia that the build process choked on building 32-bit libraries and applications on an x86_64 system under Ubuntu Fiesty (7.04). While most other distributions include the `gcc` 32-bit compatibility libraries for 64-bit architectures with a 32-bit emulation mode, Ubuntu (and maybe Debian too) does not. The Ubuntu 7.04 package that is missing is the `lib32gcc1` package. There are two workarounds to this difficulty: add the `lib32gcc1` package (with apt), or add the '`--disable-32bit-libs`' flag to '`configure`'.

  noted in `openss7-0.9.2.G`

`strnsl 002. 2007-06-20T15:22:19-0600`
: The libtirpc package for current releases of GNU/Linux including NFS4 has usurped a broken /etc/netconfig file for use by its broken implementation of TI-RPC. So when strnsl and more importantly the strinet package installs its /etc/netconfig entries, the

libtirpc package breaks. To avoid this problem, I think that the best approach is to install a libtirpc entry (or a copy of the broken /etc/netconfig) into the /etc/netconfig.d subdirectory when strnsl installs and restore it to /etc/netconfig when strnsls removes.

Fixed in the current release: the actual fix involves not calculating the /etc/netconfig file from the /etc/netconfig.d subdirectory entries when an /etc/netconfig file exists that we have not generated.

`strsctp 005. 2007-05-18T07:06:33+0000`

Found the most obnoxious of bugs. When testing on loopback, more than 255 calls were being made to `dupb(9)` causing the *db_ref* field of the data block to wrap to zero, causing an incorrect reference count.

fixed in `strsctp-0.9.2.7`

`streams 014. 2007-05-17T21:48:24+0000`

The `dupb(9)` utility had an obnoxious bug where it permitted the *db_ref* count to wrap to zero, causing buffer allocation and freeing problems. This was very difficult to debug. `dupb(9)` now fails if the reference count has reached 255. When `dupb(9)` fails, the user should check if the reference count has reached 255, and if it has, attempt a deep `copyb(9)` instead. At some point it might be useful to have STREAMS do the deep copy automatically. This was discovered in `strsctp` loopback tests where message blocks are rapidly duplicated for retransmission.

fixed in `streams-0.9.2.3`

`streams 013. 2007-05-17T21:48:06+0000`

The log driver, strace, strerr and strclean utilities had some bugs. The `strsctp` driver now makes extensive use of `strlog(9)` trace and error logging and the log driver and utilities have been corrected. These facilities are now production grade.

fixed in `streams-0.9.2.3`

`streams 012. 2007-04-13T01:47:30+0000`

It appears that *Ubuntu 6.10* has a rather broken implementation of the LSB `install_init` that has been inherited from Debian (a python script, none the less). This implementation refuses to properly install a disabled service (one with an empty or missing Default-Start: tag), but, rather invokes updated-rc.d in such a way that the init script is started at runlevels '`2 3 4 5`' instead. This was causing problems with the strace and strerr services which are normally installed disabled.

This uncovered the fact that the Debian-style init scripts were not working anyway. The scripts have been fixed and the strace and strerr utilities now default to enabled.

fixed in `streams-0.9.2.3`

`streams 011. 2007-04-10T10:56:42+0000`

The strbcflag flag was never being cleared, causing infinite looping of the scheduler once the maximum number of buffers was reached. This also revealed a problem that bufcalls were being run unncecessarily (when strbcwait was set, instead of only when strbcflag was set).

fixed in `streams-0.9.2.3`

`streams 010. 2007-04-10T10:55:29+0000`

The stream event sequence number was wrapping and becoming larger than the event mask resulting in inability to cancel buffer callbacks and timeouts.

fixed in `streams-0.9.2.3`

`strxns 002. 2007-04-02T13:04:36+0000`
: Option handling bug in `ldltest`.

  fixed in `strxns-0.9.2.6`

`streams 009. 2007-04-02T11:57:35+0000`
: `ldl` was using an incorrect MKDEV command, but when the Stream head attempted to redirect the open to the new (mangled) major device number, it properly returned ENXIO, but did not release a reference to the module. Need to check code paths for this to see where the reference needed to be released.

  known bug

`strxns 001. 2007-04-02T11:52:59+0000`
: `ldl` was not demand loading for 2.6 kernels due to missing MODULE_ALIAS declarations. Also, `ldl` had an incorrect SVR 3 style '`MKDEV(MAJOR(*devp), i)`' construct in it that was using Linux utilities instead of Linux Fast-STREAMS utilities, causing the Stream head to attempt to redirect the major device. Unfortunalely, this was failing. Changed to the correct SVR 4 style '`makedevice(getmajor(*devp), i)`'. (There must still be a bad major redirection path in the Stream head.)

  fixed in `strxns-0.9.2.6`

`streams 008. 2007-03-31T05:33:29-0600`
: When loosening SMP locking, found a bug in the QWANTR handling in `getq(9)` and back-enabling in `flushq(9)` and `flushband(9)`. Both of these were generating false back-enables. The `getq(9)` was generating a *lot* of false back-enables. Whenever `getq(9)` found an empty queue it was not only setting QWANTR, but it was back-enabling the queue. The result is that if service procedures are used exclusively (that is, `qi_put(9)` always does a `putq(9)`), `getq(9)` would generate a false back-enable for each message. Also, the enabled queue would generate another false back-enable. Significant performance gains should be noticed.

  fixed in `streams-0.9.2.3`

`streams 007. 2007-03-16T17:33:20-0600`
: Jérémy Compostella pointed out an error in `strallocpmsg(9)` where it was always assigining M_PCPROTO to messages created with I_FDINSERT.

  fixed in `streams-0.9.2.3`

`streams 006. 2007-03-14T23:48:26-0600`
: There appears to be an inode lock imbalance that occurred for several clone error paths in stropen. If the returned major device number does not correspond to a driver, or an snode cannot be acquired for the new entry and the stream head reparented.

  fixed in `streams-0.9.2.2`

`strsctp 004. 2007-03-14T17:36:31-0600`
: Another bug found, a double buffer free in `sctp_recv_msg(9)` when calling `sctp_rcv_ootb(9)`. This bug was discovered during verification testing on a high speed SMP machine.

  fixed in `strsctp-0.9.2.7`

`strss7 004. 2007-03-13T23:36:37-0600 (x400p-ss7/5766)`
: Older chips were not being recognized correctly either. Reflected device table and a few missing break statements was the cause.

  fixed in `strss7-0.9a.6`

`strss7 003. 2007-03-13T02:40:38-0600 (x400p-ss7/5766)`

It appears that V401PE cards were almost loading (configuring, mapping, downloading firmware) but were failing to configure at the final stage of the process (matching device id to board type). The device id for the 2155 is supposed to be b4 regardless of E1 or T1, but I just guess that the card is reporting 34 instead of b4 for E1 (it might be the firmware setting or unsetting the high bit as was done for the other devices). I changed the sl_x400p.c driver to print error messages when loading if such mismatches occur and set the device ids to accept 0x3X as an E1 2155 device and 0bX as a T1/J1 2155 device (even though they are all the same).

As it turns out, the V401PE cards have a DS2156 chip instead of a DS2155 chip. The only difference is the lack of Tx fixed gain control (register 0x7d) and the DS2156 supports a UTOPIA II bus that is not used on the V401PE. I changed the driver to recognize the DS2156 and skip register 0x7d when it exists. The DS2156 chip will work for both V401T (D33D) and V401E (D44D) cards.

fixed in `strss7-0.9a.6`

`strsctp 003. 2007-03-10T05:59:10-0700`

One serious locking problem discovered. `sctp_cleanup_read(9)` was suppressing IRQs across calls to `putnext(9)` when delivering data and acknowledgements. Recent kernels on Fedora and Ubuntu were complaining about IRQs suppressed across calls to `local_bh_enable(9)` in M2PA and that is what lead to the discovery.

fixed in `strsctp-0.9.2.7`

`strtty 001. 2007-03-09T15:31:23-0700`

The package was installing /dev/ptmx and /dev/pts/n device and removing them when uninstalling. This caused havoc with Linux's UNIX'95 pseudo-terminal devices. They have been strapped out until later.

fixed in `strtty-0.9.2.2`

`sigtran 001. 2007-03-08T21:09:59-0700`

The inital timeout values associated with a freshly pushed M2PA module were being set to HZ based values instead of milliseconds. This did not affect the test program (which explicitly sets the values), but could have affected applications programs on systems where the tick clock is 100 Hz.

fixed in `sigtran-0.9.2.2`

`strss7 002. 2007-03-08T21:05:47-0700`

Timers were message up (using HZ instead of milliseconds) on the X400P driver. This did not affect architectures with 1000 HZ or 1024 HZ tick clocks, but 100 HZ systems object.

fixed in `strss7-0.9a.6`

`streams 005. 2007-03-07T15:53:06-0700`

Demand loading of kernel modules for clone devices opened, for example, as /dev/streams/clone/mux was requesting module streams-clone-mux and /dev/streams/clone/mux but was not requesting streams-mux or /dev/streams/mux and the modules were failing to demand load.

fixed in `streams-0.9.2.2`

`strss7 001. 2007-03-05T15:58:14-0700`
: For some reason the package was always building and installing the `libxnet` texinfo document which was keeping the RPM from installing after the `strxnet` package. This was fixed in `doc/manual/Makefile.am` with the addition of a '`WITH_XNET`' conditional.

  fixed in `strss7-0.9a.6`

`strinet 002. 2007-03-05T01:24:13-0700`
: RPMs built on 2.4 kernels have using the '`%dev`' construct for RPM instead of installing devices using the init scripts. RPM complains loudly because the dev package defines some of the same iBCS devices (e.g. `/dev/ipip`) as we are attempting to install. In the meantime, use –force.

  Fixed in `strinet.spec` to never build devices ala rpm for strinet package.

`strnsl 001. 2007-03-05T01:21:21-0700`
: RPMs built on 2.4 kernels install the '`include strnsl`' line to `modules.conf` when there is no such directory installed by any of the rpms. This means that `depmod` complains loudly about the missing directory. This was fixed for the current release.

`streams 004. 2007-02-26T08:25:09-0700`
: Jérémy Compostella pointed out error in clone.c. When an automatic clone minor device was unregistered, it was unregistering the modid instead of the major number. This was not noticed because all OpenSS7 drivers have the same modid as major number (strconf does this automatically).

  fixed in `streams-0.9.2.2`

`streams 003. 2007-02-26T08:25:09-0700`
: Jérémy Compostella pointed out syntax error in strsched.c that kept synqs from compiling properly.

  fixed in `streams-0.9.2.2`

`strcompat 001. 2007-01-12T11:40:15-0600`
: A bug in the `mi_copyout(9)` function was discovered by inspection. The function should complete the last stage of a non-TRANSPARENT input-output control operation by returning an `M_IOCACK(9)` message, but did not. This was fixed for release `0.9.2.5`.

`strsctp test-sctp_n -o 9.1 Sun, 29 Oct 2006 16:27:35 -0700`
`test-sctp_n -o 9.2 Sun, 29 Oct 2006 16:27:35 -0700`
: When the number of test packets is set to 300, we are crashing on high speed SMP HT box. This seems to be a locking problem of sorts, or some flow control race condition. For now, the number of test packets, TEST_PACKETS has been reduced from 300 to 30 to avoid the crash. Again, packet tests at IP level should reveal this problem.

  Note that there does not seem to be a problem with similar TPI tests, so the problem might be NPI interface related after all.

`strsctp test-sctp_n -o 7.1 Sun, 15 Oct 2006 06:22:05 -0600`
: I notice that when the message size in this test is larger than the receive window size on the receiver, the receiver aborts the association after its window fills. If the message size is reduced to just beneath the receive window size, the test case succeeds. So, it looks like we are not handling zero window probes very well at all. For now I have just reduced the message size as this is for interface testing not packet testing. Packet tests at IP level should reveal this problem.

`streams 002. 2006-09-24T20:02:00+0000`

Discovered asynchronous thread cancellation inconsistencies in libLiS libpLiS by inspection during documentation. `isastream(2s)`, `fattach(2)` were not performing proper asynchronous thread cancellation suppression so that these function contained a cancellation point when the should not.

fixed in `streams-0.7a.6.rc3`

`streams 001. 2006-07-05T21:54:49+0000`

Fedora Core 5 reports a rwlock bug during udp module unloading as follows:

```
BUG: rwlock wrong CPU on CPU#0, rmmod/7515
Call Trace:
  {rwlock_bug+100}
  {_raw_write_unlock+88}
  {:streams:unregister_strnod+211}
  {:streams:unregister_clone+64}
  {:streams:unregister_strdev+24}
  {:streams_udp:udpterminate+26}
  {sys_delete_module+406}
  {system_call+126}
```

It appears that `unregister_strnod(9)` is scheduling while holding a write lock on cdevsw_lock. This is probably in `iput(9)` called within cmin_del.

fixed in `streams-0.7a.6.rc2`

There were a number of places where sleeping functions were called with spin-locks held, causing the CPU awaking from the sleep to sometimes be different from the CPU that took the lock. This was buggy, so I reworked all of these cdev and fmod sections to handle spin locks properly. FC5/SMP on HT no longer reports these bugs.

`strinet 001.`

Several test cases are failing sending messages. ICMP port unreachable errors are resulting. It appears that the caching of destination addresses is somehow making the receiver think that it is a connection-oriented stream!

7.6 Schedule

Current Plan

The *OpenSS7* package is a mature collection of all of the OpenSS7 STREAMS and protocol modules that builds all components on a wide range of supported Linux distributions and kernels. The *OpenSS7* package is a builds and validates all components against a given distribution.

Therefore, the current plan for packaging is largely a maintenance plan. The OpenSS7 Project intends to release regularly new versions of the package that build and validate against upcoming releases of the supported Linux Distributions available from major distributors and upcoming releases of the Linux kernel, both mainline and as patched by major distributions. This release schedule is approximately every 3 to 6 months. More recent corrections and support for new distributions and kernels can be obtained by sponsoring the OpenSS7 Project and obtaining access to the live CVS repository (also available as a git repository).

Two significant plans for the package include providing cross-compiling support for more cross-platform development distributions, such as the Denx ELDK. Also, support for real-time distributions such as Montavista Linux and RT releases of SuSE and RedHat are within the scope of the development plan.

No additional components are currently planned although development within the existing components are planned. See the section for the component below, for a current development plan for a specific component.

STREAMS Current Plan

There are not many things left to be done on the production Linux Fast-STREAMS component. As of the streams-0.9.3 release, performance modifications are complete. The component now exhibits performance on STREAMS-based pipes and TPI drivers that is significantly (factor of 2 or more) superior to that experienced by legacy Linux facilities.

Therefore, the current plan for Linux Fast-STREAMS is largely a maintenance plan. Items on the todo list, below, will be picked up as time permits. The OpenSS7 Project intends to release regularly new versions of Linux Fast-STREAMS that build and validate against upcoming releases of the supported Linux Distributions available from major distributors and upcoming releases of the Linux kernel, both mainline and as patched by major distributors. This release schedule is approximately every 3 to 6 months. More recent corrections and support for new distributions and kernels can be obtained by sponsoring the OpenSS7 Project and obtaining access to the live CVS repository (also available as a git repository).

One development activity in the works for Linux Fast-STREAMS is to provide integral support for more embedded cross-platform development systems such as the Denx ELDK, as well a existing and emerging RT kernels such as Montavista and the upcoming SuSE and RedHat RT kernels. This is a significant undertaking and will only be embarked upon when the OpenSS7 Project is given free access to these RT kernels and distributions.

COMPAT Current Plan

There are not many things left to be done on the production OpenSS7 STREAMS Compatibility Modules component. The current plan for the component is largely a maintenance plan including support for current distributions and kernels.

There are currently a large array of Solaris DDI compatibility functions that are not implemented; however, there have not been many requests for this capability. Perhaps the advent of OpenSolaris has forstalled porting of many drivers to Linux, but, regardless of the cause, there is just not a demand. If there are any functions that you need the OpenSS7 Project to support that are not currently supported in one of the compatibility modules, please request support for them on the openss7-develop mailing list.

BCM Current Plan

The OpenSS7 Project has made several stabs at making this component available and providing it in a production grade form. All attempts are currently incomplete. All in all there does not appear to be sufficient interest in this capability to actually fund the work. Therefore, this component will remain incomplete until some entity can justify funding the remainder of the development. The OpenSS7 Project remains committed to the open source model and providing this support runs somewhat against that. However, if your organization has a pressing need for this capability and can offer funding for its completion, please contact the project on the openss7 mailing list.

TTY Current Plan

This component is still incomplete. It is missing a fully functional ldterm module. Also, performance testing of STREAMS-based pipes from the Linux Fast-STREAMS indicates that STREAMS-based pseudo-terminals could exhibit far superior performance to that exhibited by the legacy Linux (SVR

3 style) pseduo-terminals. To accomplish this requires fully implementing ldterm, creating validation test suites, and peformance analysis and comparison with legacy Linux mechanisms.

This component 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 component is important in your industry and your organization is able to fund further development or contribute the missing items, contact us on the openss7-develop mailing list. Until funding or a mandate surfaces, this component will likely continue as a proof-of-concept only. It will be maintained in a compilable and installable state (that is, it will be updated for current Linux distributions and kernels) on the same basis as other components in the OpenSS7 package.

CHAN Current Plan

This component, which contains hard-switch or MG capabilities is not currently planned. Of priority is the production deployment of soft-switch or MGC and Signalling Gateway components from the `SIGTRAN`, `SS7` and `VOIP` components. This component will only receive maintenance releases until those priorities have been sufficiently met (over the course of Summer 2009).

XNS Current Plan

This component is mature and complete and in production release. The current plan is to provide only maintenance releases supporting more recent Linux distributions, kernels and tool chains as they evolve. Expect public maintenance releases on a 3 to 6 month cycle.

XNET Current Plan

This component is mature and complete and in production release. The current plan is to provide only maintenance releases supporting more recent Linux distributions, kernels and tool chains as they evolve. Expect public maintenance releases on a 3 to 6 month cycle.

NSL Current Plan

Current plans are to just maintain this component for recent distributions, kernels and tool chains. Although at some point the relation of this component to TI-RPC 2.8 needs to be worked out, when that will happen is not currently on a schedule. I do not really know how much of a requirement there is for ONC RPC running over other transports such as ISO transports and SCTP transports, although it would surely be interesting to run NFS4 over SCTP.

This component 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 component is important in your industry and your organization is able to fund further development or contribute the missing items, contact us on the openss7-develop mailing list. Until funding or a mandate surfaces, this component will likely continue as a proof-of-concept only. It will be maintained in a compilable and installable component (that is, it will be updated for current Linux distributions and kernels) on the same basis as other components in the OpenSS7 package.

SOCK Current Plan

This component 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.

This component 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 component 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 component will likely continue as a proof-of-concept only. It will be maintained ina compilable and installable component (that is, it will be updated for current Linux distributions and kernels) on the same basis as other components in the OpenSS7 package.

INET Current Plan

This component is mature and complete and in production release. The current plan is to provide only maintenance releases supporting more recent Linux distributions and kernels as they evolve. Expect public maintenance releases on a 3 to 6 month cycle.

SCTP Current Plan

This component is mature and complete and in production release. The current plan is to provide only maintenance releases supporting more recent Linux distributions, kernels and tool chains as they evolve. Expect public maintenance releases on a 3 to 6 month cycle.

X25 Current Plan

This component was initially started to provide managment interfaces to platforms using the OpenSS7 protocol stacks (CMIP and CMOT interfaces) using SS7 GDMO and other GDMO. It was considered pulling some of the old `isode` package into this component to provide some GDMO facilities.

Since then, it was discovered that these ISO protocols are still very important to a number of industries on Linux and Linux Fast-STREAMS. These industries are the Aviation and Financial industries. Closer to the heard of telecommunications is the applications of aircraft to ground communications for the Aviation industry.

While the OpenSS7 Project's focus over the Summer of 2009 is going to be soft-switch enabling protocols, that may change if the Aviation industry steps up to this component with some funding to complete the work or by contributing changes, improvements or development to this component. Otherwise, expect this component to receive only maintenance releases over the next 8 months or so.

ISO Current Plan

This component was initially started to provide managment interfaces to platforms using the OpenSS7 protocol stacks (CMIP and CMOT interfaces) using SS7 GDMO and other GDMO. It was considered pulling some of the old `isode` package into this component to provide some GDMO facilities.

Since then, it was discovered that these ISO protocols are still very important to a number of industries on Linux and Linux Fast-STREAMS. These industries are the Aviation and Financial industries. Closer to the heard of telecommunications is the applications of aircraft to ground communications for the Aviation industry.

While the OpenSS7 Project's focus over the Summer of 2009 is going to be soft-switch enabling protocols, that may change if the Aviation industry steps up to this component with some funding to complete the work or by contributing changes, improvements or development to this component. Otherwise, expect this component to receive only maintenance releases over the next 8 months or so.

ISDN Current Plan

This component is lower down on the priority list. Current focus is on pushing through the `SIGTRAN`, `SS7` and `VOIP` components instead. This components sits at about the same status as the `CHAN` component. there exists modules and drivers in this component that have not yet been publicly released and are only available to sponsors of the OpenSS7 Project on the CVS archive. When the project gets back to this component (sometime 2H2009), the plan will be to move the private modules and drivers into the public release, place conformance validation test cases around them and release this component as production grade.

SS7 Current Plan

The current plan includes the public production release of SCCP, ISUP and TCAP components. MTP3 has not yet been publicly released and is not currently scheduled. (Implementing the AS-side of M3UA is taking precedence.) The first public production releases of SCCP, ISUP and TCAP are planned for the end of July 2009. This will be followed by lab and limitied field trial testing with a lead customer in August and production deployment in the lead customer's network in 1Q2009. Expect that over the next 8 months releases of this component will include major new development and validation testing as well as including SCCP, ISUP and TCAP components that have nor heretofore been publicly released.

SIGTRAN Current Plan

The current plan of the next several months (over the summer of 2009) is to actively move this component forward. This consists of the following:

- Move SIGTRAN modules and drivers that have not yet been publicly released into the public release.
- Write test cases for complete AS-side implementations of all of the UAs (including ISUA and TUA).
- Perform production testing of the UAs against Tekelec Eagle and Cisco ITP equipment.
- Rigorous lab and limited field testing of AS-side components.
- Production deployment in customer's network for 1Q2009.

Note that M2PA is complete. Aside from that, the order of testing and implementation will be M3UA followed by M2UA or SUA and then TUA, IUA (including V5UA and GR303UA) and ISUA.
Production public releases of the AS side of M3UA can be expected by the end of July 2009.

ATM Current Plan

This component was initially started to provide managment interfaces to platforms using the OpenSS7 protocol stacks (CMIP and CMOT interfaces) using SS7 GDMO and other GDMO. It was considered pulling some of the old `isode` package into this component to provide some GDMO facilities.
Since then, it was discovered that these ISO protocols are still very important to a number of industries on Linux and Linux Fast-STREAMS. These industries are the Aviation and Financial industries. Closer to the heard of telecommunications is the applications of aircraft to ground communications for the Aviation industry.
While the OpenSS7 Project's focus over the Summer of 2009 is going to be soft-switch enabling protocols, that may change if the Aviation industry steps up to this component with some funding to complete the work or by contributing changes, improvements or development to this component. Otherwise, expect this component to receive only maintenance releases over the next 8 months or so.

VOIP Current Plan

The current plan of the next several months (over the summer of 2009) is to actively move this component forward. This consists of the following:

- Implementation and testing of MGCP (MGC-side) using the MGI and TCP transport from the `INET` component.
- Implementation and testing of MEGACO/H.248 (MGC-side, binary format) using the MGI and SCTP transport from the `SCTP` component.
- Implementation of SIP-T and SIP call control using the CCI and TCP transport from the `INET` component and later SCTP transport from the `SCTP` component.
- Perform production testing of MEGACO/H.248 and MGCP against the Veraz iGATE, the Cisco 5400 and the Sonus Media Gateways.
- Rigorous lab and limited field testing of the MGC-side components.
- Production deployment in customer's network for 1Q2009.

The order of testing will be MEGACO/H.248 and then SIP-T and SIP followed by MGCP. Production public releases of the MGC side of MEGACO/H.248 and SIP-T acan be expected by the end of July 2009.

Things to Do

– Packaging.

  Get master build package working better. Currently the master build package does not build a master tarball or RPM distribution, which would be nice, only individual RPMs for subpackages.

– Cross-testing framework.

  We have nice autotest test suites that are fine for native builds, but for cross-compile builds, it would also be nice to cross-test. We can accomplish this nicely within the autoconf framework using DejaGNU. DejaGNU configured under the STREAMS, but we need a more general autoconf .m4 fragement check for the existence of DejaGNU, and automake and DejaGNU expect wrappers for the existing test programs (and integration into the make check-DEJAGNU target). Then we could cross-build the package and then execute the test scripts on the result on a remote board using DEJAGNU.

– Distributed testing framework.

  Because these are mostly communications protocols implemented using STREAMS, it is important to be able to do consistent distributed testing and validation of the protocol implementations. We can perform much validation using Ferry-Clip approaches (linking or pushing STREAMS-based pipes beneath drivers or modules), however, acceptance and performance testing would benefit from a distributed framework. Perhaps the most direct yet general approach to this is TETware from OpenGroup and is used for POSIX (and XNS 5.2, by the way) test suites from OpenGroup. Although netperf incorporates its own client-server subsystem, TETware provides these mechanisms separately. TETware also provides mechanisms for test case synchronization points between distributed systems that makes distributed conformance test cases quite possible. Note that TETware does not really require any external tools but does require network access and installation on the target systems (unlike DejaGNU that can execute test cases over a serial port if necessary).

– Dynamic configuration.

  Automake files (Makefile.am) are too static. The strbcm package needs the list of sources and objects to be rather dynamic. It would be nice to convert all packages to the same approach.

We could perhaps use strconf-sh to generate Makefile.in fragments at configuration time and then include them in a far more general main Makefile.am file.

– Merge sub-packages.

I really want to put all these packages back together, it is too time consuming maintaining the various administrative files for each of seven or so packages.

- streams-0.7a.5
- strcompat-0.9.2.4
- strutil-0.9.2.4
- strxns-0.9.2.4
- strxnet-0.9.2.9
- strinet-0.9.2.4
- strsctp-0.9.2.6
- netperf-2.3.2

The result will be just a streams-0.9.4 release. The only two (STREAMS) packages left will be strbcm and strss7.

STREAMS Things to Do

– Support for RT kernels. This is a little more than just having the STREAMS scheduler run as a non-RT process kernel thread, which it does now, and which is trivial. (The existing package should compile and run against these kernels with minor modification in this event.)

More to the point is working the light-weight STREAMS scheduler and service procedures into a prioritized scheme where service procedures run as real-time, yet pre-emptable tasks. In contrast to the current scheme, it is likely that the approach would be to either spawn multiple kernel threads for the STREAMS scheduler at different priorities, or to alter the priority of the STREAMS scheduler in response to the scheduling of specific queues at specific priorities. A design is not really possible until the intricacies of upcoming RT kernels are discovered.

TODO: Provide support for RT kernels.

– Per cpu data:- I am still using the older approach of using cache line aligned arrays for per-cpu data. This, of course, does not fully utilize NUMA architectures. For NUMA architectures we need to use the per-cpu utilities provided by the 2.6 and 3.x kernel. I haven't touched converting this yet.

Also, there are several NUMA supporting STREAMS utility functions (allocb_node, etc.) that need to be supported yet.

TODO: Convert cacheline aligned arrays to NUMA per-cpu data on 2.6 and 3.x kernels. Complete NUMA supporting STREAMS facilities.

– Provide support for assigning a processor affinity for queue pairs. The current STREAMS scheduler will schedule a queue enabled as a result of, for example, a put() on the same processor that performed the action that caused the enable. This has been adequate for event driven systems. However, for pipelined hard real-time, better processor instruction cache efficiency and concurrency might be gained by assigning portions of the pipeline to different processors, so that, for example, when a put is performed to a queue, that the queue will become enabled against the process with which it has an affinity rather than the enabling processor. This might permit assigning a different processor affinity to each queue-pair in a pipeline to exploit concurrency in the pipeline.

TODO: Add an optional processor affinity to the STREAMS scheduler.

– Split `include/sys/streams/stropts.h` by architecture. There is conflicting numbering on the standard STREAMS input-output controls:

```
I_SWROPT(7)       I_GWROPT(7)       I_LIST(7)
I_FLUSHBAND(7)    I_CKBAND(7)       I_GETBAND(7)
I_ATMARK(7)       I_SETCLTIME(7)    I_GETCLTIME(7)
I_CANPUT(7)
```

System V Release 4 UNIX® vendors use one set and *OSF* UNIX® vendors use another. Namely *HP-UX*, *OSF/1.2*, *AIX*, *Mac OpenTransport* use OSF numbering, whereas *IRIX*, *Solaris*, *UnixWare* and others use SVR4 numbering. So, for HPPA, Alpha, PowerPC, we should use the OSF numbering.

I know that it is a fall-back to the SVR4 way of separating architectural differences by UNIX vendor (if it is HPPA, it must be sold by HP and it must be HP-UX running on it, for example), but even the Linux kernel is victim to this (many ioctls and some errno numbering is split this way). It is completely entrenched in GNU autoconf's config.guess.

TODO: Split `include/sys/streams/stropts.h` by processor architecture.

– A similar numbering mismatch occurs for many of the message block types.

TODO: Split `include/sys/streams/streams.h` by processor architecture.

– Implement `I_EGETSIG(7)` and `I_ESETSIG(7)`. These are *Solaris* enhanced version of the `I_GETSIG(7)` and `I_SETSIG(7)` STREAMS input-output controls. The difficulty with their implementation is that the entire signal handling setup inside the Stream head code is geared toward the calling process and needs to be adjusted to be general enough for any process or process group. Until then, *Linux* file asynchronous I/O is supported.

PARTLY DONE:
Wrote the manual pages and added them to the build. Placed function skeletons that return `[EOPNOTSUPP]` for these functions in the Stream head.

TODO: Implement `I_EGETSIG(7)` and `I_ESETSIG(7)`.

– Socket buffer handling:

1. Rather than write offset and padding, why not provide a flag (e.g. SO_SKBUFF) to indicate to the stream head to allocate an sk_buff with the message block and share buffers between mblk and sk_buff, then, the sk_buff can be used without allocation in the bottom half. esballoc() and alloc_skbuff() can be used to set up the message block. dup() could be made aware of the hidden sk_buff and increment the shared sk_buff count as well. Also, msgpullup() and pullupmsg() could be made aware of message blocks containing sk_buffs and have them do the appropriate thing.
2. The other thing that is needed is some way to tell the other end of a loopback connection that the sk_buff it has received already has an mblk attached to it as above. Then the message block could be simply passed upstream and one would not need to be esballoc'ed for it.
3. Another thing is to provide the ability to partial checksum and copy data from user into these sk_buffs, but setting an SO_CSUM flag along with the SO_SKBUFF flag to indicate the type of checksum to perform.

The combination of the above three items should provide some serious performance gains for Linux networking based stream heads.

PARTIALLY DONE:
Item (1) is done and complete. The 2nd generation UDP and RAW drivers are already using it. Item (2) and (3) remain.

– Had another look at specfs, devfs and udev. It looks like we can create minor device nodes within /dev (not just /dev/streams) using devfs or udev. Again, this doesn't do everything that specfs does. specfs will demand load when an attempt is made to open a non-existent character device. Nevertheless, we can describe a "streams" class for udev and when a module registers a minor device node, we can have udev create that device node and provide permissions by adding our files to the /etc/udev/rules.d and /etc/udev/permissions.d directories.

  Therefore, on a udev system, we should make strconf-sh create the necessary rules.d and permissions.d file entries. register_strnod will be modified to create a udev instance within the stream class matching the rules.d and permissions.d entry when creating a minor device node within the specfs.

  On a devfs system, register_strdev and register_strnod should perform devfs calls instead of calling register_chrdev. That way minor device nodes will automatically appear at least once the module is loaded.

  TODO: rationalize specfs to devfs and udev

– Have the STREAMS subsystem register a panic notifier on 2.6 and 3.x kernels to be able to recover from panics caused by misbehaving STREAMS modules or drivers.

  TODO: Register panic notifier.

– Kernel objects are another thing. For 2.6 or 3.x kernels, we need to hold our data structures in the kobject manner so that the /sys file system is usable. This requires another adaptation layer because 2.4 kernels do this in a completely different way. Much of our /proc file system stuff needs to move into /sys for 2.6 or 3.x kernels but stay the same for 2.4 kernels.

  The /sys file system does not really do much for STREAMS. The /dev/streams specfs file system does more for us.

  SKIPPED.

COMPAT Things to Do

– Implement the AIX strtune() command.

  todo

– I would really really like a set of rmallocmap(), rmalloc(), rmalloc_wait(), rmfree(), rmfreemap() functions so that drivers could stop using the kmem_cache functions but could acheive similar effect.

  todo

– Work in MUTEX_ALLOC(), MUTEX_DESTROY(), MUTEX_LOCK(), MUTEX_MINE(), MUTEX_OWNED(), MUTEX_TRYLOCK(), MUTEX_UNLOCK() from IRIX into irixcompat.c.

  todo

– Work in streams_interrupt() and STREAMS_TIMEOUT() from IRIX into irixcompat.c.

  todo

– Hey, here's an idea for testing Solaris compatibility: take an OpenSolaris source file for a STREAMS driver and compile and test it under Linux with no (or minimal) source code modifications!

  todo

- Write test programs and test suites. There are really not any test programs or test suites available for the OpenSS7 STREAMS Compatibility Modules package as of strcompat-0.9.2.2.

  todo

- Not all compatibility functionality is implemented. There are a large number of Sun DDI functions applicable to STREAMS that have not been fully implemented. Also, the Sun configuration management mechanism is not yet fully implemented and neither is that for AIX. Also, there is a large group of SVR 4.2 compatible functions that are not directly STREAMS related but are part of the DDI/DKI and should be implemented to provide abstraction from Linux internals as well as the ability to link binary modules.

  moved (Note that the ability to link binary modules has been moved to is own `strbcm` package.)

- Linking of binary modules is not yet supported. This is the place (strcompat) where binary modules should be permitted to be loaded against Linux Fast-STREAMS, because the binary compatibility interface modules are defined here. There is the beginnings of an strconf script output to generate a C-language wrapper file that will link with a binary object file to generate a loadable module that could load under Linux Fast-STREAMS.

  moved (Note that the ability to link binary modules has been moved to is own `strbcm` package.)

- Documentation. The documentation is trailing a bit. I have thousands of manual pages written, however, some are sparse or incomplete. Also, the manual and the STREAMS Porting Guide needs a bunch of work.

  todo Did a bunch of work on the manual, however, there is still a lot of work on a STREAMS Porting Guide to be done.

BCM Things to Do

- Testing. This package is completely untested. One way to test compatibility is to take a set of frozen binary modules and drivers from the release and place them in a test directory.
- The strbcm package is currently incomplete. The purpose of the package was to move STREAMS binary compatibility module capabilities outside of the STREAMS release package. As Linux Fast-STREAMS has become production grade 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.

  This package; however, might has some use still outside of Linux Fast-STREAMS if only because it provides a separable way of incorporating 3rd party binary modules.

TTY Things to Do

- Testing. This package is completely untested.

  todo

- Create a library `libtty(3)` to hold the library functions for use with this package:

  `openpt(3)`
  `grantpt(3)`
  `ptsname(3)`
  `unlockpt(3)`

  todo

– Create an `ldterm(4)` *Line Discipline for Terminals* STREAMS module. This item remains to be done. Al that is in place right now is a skeleton module.

  todo

The `openss7` package is currently incomplete.

The purpose of the package was to provide STREAMS terminal capabilities for *Linux Fast-STREAMS*. The package will only build and install with *Linux Fast-STREAMS*

If someone is interested in this package, a contribution of a working `ldterm(4)` module would be good. Also, testing could be performed.

CHAN Things to Do

– Testing. This package is completely untested.

  This remains to be done.

– Actually implement all of the package.

– Copy all of the channel drivers, multiplexers and switching drivers as well as media gateway drivers from the strss7 package and place it here.

  Note that this package should load before the strxns package because it provides channels that can be used for hdlc links and data links and the strxns package provides the CDI and DLPI interfaces. It is possible that the CDI interface should be moved here.

XNS Things to Do

– I want to write a NetFilter pseudo-device driver that will install and control netfilter hooks for performance testing. It will do things like packet drops, checksum errors, packet duplication, packet delay, and combinations, for a specified range of addresses and/or protocols. Also a set of user-space utilities (C language programs) to permit the control of the filters. The purpose is to be able to do a wide range of performance testing on simulated networks that are simply loop back devices. This would permit performance test programs (such as netperf) to be used for testing special network conditions on SCTP (for example), and will also allow for comparison tests.

• The strxns package contains several GPL'd modules and drivers originally included in the STREAMS releases that were removed to this package so that they could be used with Linux Fast-STREAMS.

– Test. Almost none of the modules or drivers in this package have been tested thoroughly against Linux Fast-STREAMS. One of the primary reasons for this is that The OpenSS7 Project does not really use any of these modules or drivers.

– This might be a useful place to drop all of the DLPI drivers from the strss7 package, particularly those for X.25, Frame Relay, and ISDN LAPD and LAPB.

XNET Things to Do

• The texinfo manual, `libxnet.texi`, is incomplete. For the time being, the manual pages, starting with `libxnet(3)`, are the definitive documentation. It would be nice to also provide these library manual pages as a printable manual, however, the project has not yet found the time to complete this manual.

  todo

• There is an XTITRACE option the was associated wtih the XTI library that provided the abilty to record trace information to a temporary file and then display or print it with an xtitrace

user program. This capability is not yet implemented in this XTI library and remains to be done.

todo

- Need to remove the following functions: `t_sndopt(3)`, `t_sndvopt(3)`, `t_rcvopt(3)` and `t_rcvvopt(3)`; as they are non-standard and non-portable. Instead use the following functions: `t_sndudata(3)`, `t_sndvudata(3)`, `t_rcvudata(3)` and `t_rcvvudata(3)`; because they have precisely the same argument templates as the other. These functions are described in the Open-Group documentation as inapplicable to connection-oriented transports, however, we could relieve that restriction for SCTP. The characteristics to check is if T_CLTS, okay, T_COTS or T_COTS_ORD, only okay if the tsdu_size is not T_INVALID (as it is for TCP). (It would be okay to be T_INFINITE as it is for SCTP.)

  todo

  They may have already been removed. Verify this. They certainly have to be implemented this way yet.

- The XTI/TLI library is now largely in maintenance mode. Not much special development is planned.

– Full ATM Support. This is an item where the XTI implementation is lacking. When we provide full TPI ATM support under the strss7 package, then this item will be investigated.

todo

NSL Things to Do

- Fedora 7 has a libtirpc library that uses the /etc/netconfig files. To avoid conflicts, the installation scripts now avoid overwriting this file. An interesting thing is that the libtirpc implementation is taken from TI-RPC 2.3 when there is a TI-RPC 2.8 available. In generating an strrpc package, it should replace this broken libtirpc library with a TI-RPC 2.8 library supporting STREAMS using this strnsl package.

  todo

- The package still needs to be internationalized.

  todo

- This implementation of the Network Selection facility and Name-to-Address mapping allows direct use of TI-RPC for Linux. The primary benefit of this is the ability to use TI-RPC code directly without significant and thus support upper level services such as NFS Version 4. Another amazing possibility is to not only immediately run NFS Version 4, but also to run it over SCTP. A todo item is to take the TI-RPC 2.8 release and port it to Linux Fast-STREAMS. This might best be approached with a separate strrpc package, or addressed directly by this package.

  todo

- The texinfo manual, `libxnsl.texi`, is incomplete. For the time being, the manual pages, starting with `libxnsl(3)`, are the definitive documentation. It would be nice to also provide these library manual pages as a printable manual, however, the project has not yet found the time to complete this manual.

  todo

- This package is largely untested. A conformance test suite and some test cases need to be written.

  todo

SOCK 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

The `openss7` package is currently incomplete.
The purpose of the package was to move STREAMS socket capabilities outside of the STREAMS release package. As *Linux Fast-STREAMS* has already shown production grade 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*.

INET Things to Do

– Loop back devices (ticlts, ticots, ticotsord) are currently implemented in the `inet(4)` driver using UNIX domain sockets and the XTI over sockets approach. It should be straightforward to implement these loopback devices directly in STREAMS without involving sockets. These second generation loopback drivers need to be written yet.
todo

– Implement pseudo-connection oriented modes in INET(4) driver as well as udp2(4) and rawip2(4) drivers. Document use of pseudo-connection oriented modes for UDP and RAWIP. Pseudo-connection oriented modes permit a connectionless (T_CLTS) provider such as udp to provide the appearance of connection oriented service. This makes conversion from the Sockets API to XTI API easier.
Note that this is also very applicable to MTP (quasi-associated signalling such as ISUP) and SCCP protocol class 0 and 1 and TCAP.
todo

– It might be worth retrying the netperf(1) performance test on the second generation UDP driver with this pseudo-connection oriented mode in place. You see, the netperf tests for UDP sockets performs a connect(3).
todo

– Implement pseudo-connectionless modes in INET(4) driver as well as sctp(4) drivers. Document use of pseudo-connectionless modes for SCTP (not really applicable to TCP). Pseudo-connectionless modes permit a connection-oriented (T_COTS) provider such as sctp to provide the appearance of connectionless service. This makes support for the SCTP one-to-many model

easier. Also, it provides a way (t_sndudata(3), t_sndvudata(3), t_rcvudata(3), t_rcvvudata(3)) of passing options associated with the packet to the provider either in a T_OPTDATA or in a T_UNITDATA. This would make support of SCTP's many per-packet options more workable for XTI as well as providing a easier translation between the sockets API and XTI API for sctp.

Note that this is also very applicable to SCCP protocol classes 2 and 3 and BSSAP.

todo

- Convert inet driver to use os7 common functions. In fact, it might be a better idea to convert the driver to use MPS common functions instead as we are now trying to move away from os7 common functions.

  todo

- Connectionless testing using netperf competes well with connection-oriented sockets. Need to still modify netperf to use (pseudo-)connection-oriented UDP for better comparison.

  todo

SCTP Things to Do

- The strsctp package is quite stable and mature as of strsctp-0.9.2.2 and most of the things that remain to be done are rigorous conformance and performance testing.

  todo

- Move the tpiperf(4) module to the strxnet package and rename it tiperf(4).

  todo

- Implement pseudo-connectionless modes in sctp(4) driver. Document use of pseudo-connectionless modes for SCTP. Pseudo-connectionless modes permit a connection-oriented (T_COTS) provider such as sctp_t(4) to provide the appearance of connectionless service. This makes support for the SCTP one-to-many model easier. Also, it provides a way (t_sndudata(3), t_sndvudata(3), t_rcvudata(3), t_rcvvudata(3)) of passing options associated with the packet to the provider either in a T_OPTDATA or in a T_UNITDATA. This would make support of SCTP's many per-packet options more workable for XTI as well as providing an easier translation between the sockets API and XTI API for sctp.

  Note that this is also very applicable to SCTP protocol classes 2 and 3 and BSSAP.

  Perhaps the first proper step for this is to rewrite t_sndudata(3), t_sndvudate(3), t_rcvudata(3), and t_rcvvudata(3) handle T_OPTDATA_REQ(7) and T_OPTDATA_IND(7) primitives when in T_COTS mode.

  todo

- Work recent SCTP I-G changes into the code. Many of the I-G changes over the years have been implemented in the code, however, some of the more recent changes that made it into RFC 4460 have not yet been added. This is an ongoing process.

  todo

- Writing new IP-packet-level conformance test suites.

  todo

- It would also be an idea to make the STREAMS SCTP Driver able to be pushed as a module over a STREAMS-based pipe end so that Ferry-Clip conformance testing could be performed. In fact, Ferry-Clip testing might be better and more flexible than IP-packet-level testing because it is possible to closely control the timing of arriving packets on a STREAMS-based pipe. In

fact, this is such a good idea, I think that I will proceed along the Ferry-Clip lines for testing first and only perform the IP packet-level testing later.

todo

X25 Things to Do

- Important drivers and modules for the financial industry (POS, ATM, EFT) in support of POS to data center communications and data center to branch communications is as follows:

  X.25

  `xot(4)`

- Important drivers and modules for the aviation industry (ATN) in support of air-ground ground stations and Boundary Intermediate System and Intermediate System ground-ground stations are as follows:

  ISO 8208 SNDCF
  ISO 8802 SNDCF
  Mobile 8208 SNDCF
  CIDIN SNDCF

  `clnp(4)` With security adaptations for the ATN network.

  `esis(4)` With subset for the ATN network.

  `isis(4)` With subset for the ATN network.

  `idrp(4)` With subset for the ATN network.

  Additional drivers and modules in support of ground-to-ground Intermediate systems and End Systems:

  `isot(4)` RFC 1006, ISO Transport over TCP.

  `itot(4)` RFC 2126, ISO Transport over TCP.

  `xot(4)` RFC 1613, Cisco X.25 over TCP.

  Additional drivers and modules in support of ground end systems:

  `tp(4)`

- Testing. This package is completely untested.

  todo

- Create `isot(4)`, `itot(4)`, `lpp(4)` and `cmot(4)` STREAMS modules and drivers.

  todo

The purpose of the package was to move STREAMS ISO networking capabilities outside of the `strss7` release package. The `openss7` package is currently incomplete. If you are interested in the completion of this add-on package, contact `info@openss7.com`.

ISO Things to Do

- Important drivers and modules for the financial industry (POS, ATM, EFT) in support of POS to data center communications and data center to branch communications is as follows:

  X.25

  `xot(4)`

– Important drivers and modules for the aviation industry (ATN) in support of air-ground ground stations and Boundary Intermediate System and Intermediate System ground-ground stations are as follows:

  ISO 8208 SNDCF
  ISO 8802 SNDCF
  Mobile 8208 SNDCF
  CIDIN SNDCF

  `clnp(4)` With security adaptations for the ATN network.

  `esis(4)` With subset for the ATN network.

  `isis(4)` With subset for the ATN network.

  `idrp(4)` With subset for the ATN network.

  Additional drivers and modules in support of ground-to-ground Intermediate systems and End Systems:

  `isot(4)` RFC 1006, ISO Transport over TCP.

  `itot(4)` RFC 2126, ISO Transport over TCP.

  `xot(4)` RFC 1613, Cisco X.25 over TCP.

  Additional drivers and modules in support of ground end systems:

  `tp(4)`
– Testing. This package is completely untested.

  todo
– Create `isot(4)`, `itot(4)`, `lpp(4)` and `cmot(4)` STREAMS modules and drivers.

  todo

The purpose of the package was to move STREAMS ISO networking capabilities outside of the `strss7` release package. The `openss7` package is currently incomplete. If you are interested in the completion of this add-on package, contact `info@openss7.com`.

ISDN Things to Do

– Testing. This package is completely untested.

  This remains to be done.
– Actually implement all of the package.

SS7 Things to Do

There is a great long list of these, but here is a crack at the first ones:

– Compile entire (private) package against Linux Fast-STREAMS.
– Convert test programs into conformance test suites under GNU autotest. This should be straight forward matter.
– Start retesting SIGTRAN components against SCTP and Linux Fast-STREAMS starting with M2PA.
– Start retesting SS7 Device Drivers under Linux Fast-STREAMS starting with X400P.
– Start retesting SS7 Stack components (MTP2, MTP3, SCCP, ISUP, TCAP) under Linux Fast-STREAMS.

- Package a public release. It has been way too long since a public release of this package. The package has almost a million lines of code in it, most of which the public has never seen!
- Complete the SL-MUX driver and utility programs.
- Complete the libslpcap library (`pcap(3)`) for live capture of SS7 link SDUs using `ethereal(1)` or `wireshark(1)`.

SIGTRAN Things to Do

- Testing. This package is completely untested.

  This remains to be done.
- Actually implement all of the package.

ATM Things to Do

- Important drivers and modules for the financial industry (POS, ATM, EFT) in support of POS to data center communications and data center to branch communications is as follows:

  X.25

  `xot(4)`
- Important drivers and modules for the aviation industry (ATN) in support of air-ground ground stations and Boundary Intermediate System and Intermediate System ground-ground stations are as follows:

  ISO 8208 SNDCF
  ISO 8802 SNDCF
  Mobile 8208 SNDCF
  CIDIN SNDCF

  `clnp(4)` With security adaptations for the ATN network.

  `esis(4)` With subset for the ATN network.

  `isis(4)` With subset for the ATN network.

  `idrp(4)` With subset for the ATN network.

  Additional drivers and modules in support of ground-to-ground Intermediate systems and End Systems:

  `isot(4)` RFC 1006, ISO Transport over TCP.

  `itot(4)` RFC 2126, ISO Transport over TCP.

  `xot(4)` RFC 1613, Cisco X.25 over TCP.

  Additional drivers and modules in support of ground end systems:

  `tp(4)`
- Testing. This package is completely untested.

  todo
- Create `isot(4)`, `itot(4)`, `lpp(4)` and `cmot(4)` STREAMS modules and drivers.

  todo

The purpose of the package was to move STREAMS ISO networking capabilities outside of the `strss7` release package. The `openss7` package is currently incomplete. If you are interested in the completion of this add-on package, contact `info@openss7.com`.

VOIP Things to Do

- Testing. This package is completely untested. This remains to be done.

  todo
- Actually implement all of the package.

  todo
- Copy all of the VoIP drivers, multiplexers and modules as well as the test programs from the strss7 package and place them here.

  Note that this package should load after strss7 (i.e, last).

  todo

7.7 History

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

8 Installation

Please note that the installation instructions have changed. Previously OpenSS7 releases consisted of a number of subpackages to a master package: this is no longer the case. All OpenSS7 release software is now contained in a single package. This change was necessary to reduce the maintenance burden cause by a growing number of subpackages. The net benefit of this changes is that the resulting tarballs, RPMs and DEBs are far less error prone due to far fewer dependencies between RPMs and DEBs.

8.1 Repositories

The OpenSS7 package release can be accessed from the repositories of The OpenSS7 Project. For `rpm(8)` based systems, the package is available in a `yum(8)` repository based on `repo-md` XML and may also be accessed using `up2date(8)`, `zypper(8)` or `yast(8)`. For `yast(8)` based systems, the package is also available as a `yast(8)` installation source. For `dpkg(1)` based systems, the package is available in an `apt(8)` repository.

By far the easiest (most repeatable and manageable) form for installing and using the *OpenSS7* packages is to install packages from the `yum(8)`, `yast(8)` or `apt(8)` repositories. If your distribution does not support `yum(8)`, `zypper(8)`, `up2date(8)`, `yast(8)` or `apt(8)`, then it is still possible to install the RPMs or DEBs from the repositories using `rpm(8)`, `dpkg(1)`; or by using `wget(1)` and then installing them from RPM or DEB using `rpm(8)` or `dpkg(1)` locally.

If binaries are not available for your distribution or specific kernel, but your distribution supports `rpm(8)` 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(8)` or `dpkg(1)`.

If your architecture does not support `rpm(8)` or `dpkg(1)` at all, or you have special needs (such as cross-compiling for embedded targets or for development), the final resort method is to download, configure, build and install from the tarball.

8.1.1 Repository Access

Repositories are located under `https://www.openss7.org/repo`. The path from there to the specific repository consists of the following components:

packaging For `yum(8)` and `yast(8)` repositories, this is always `rpms`. For `apt(1)` repositories, this is always `debs`. For `tar(1)` repositories, this is `tarballs`.

distro The identifier of the distrbution. Example, `centos`, `debian`.

release The release of the distribution. Example, `5.5`, `squeeze`.

arch The architecture of the release. Example , `x86_64`, `amd64`.

So, for example, the *CentOS 5.5* distribution for `x86_64` is located under `https://www.openss7.org/repo/rpms/centos/5.5/x86_64/`; the *Debian Squeeze* distribution for `i386` is located under `https://www.openss7.org/repo/debs/debian/squeeze/i386/`; the tarball distribution is located under `https://www.openss7.org/repo/tarballs`.

Note that repository access is restricted. You may download the `repo` packages without restriction, however, when installing the `repo` package you will be prompted for your registration username and repository password. The RPMs or DEBs that you will be allowed access to will depend on your entitlement.

For the following distributions, follow the instructions in the sections referenced:

CentOS Enterprise Linux 4.92	centos49	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 5.0	centos5	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 5.1	centos51	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 5.2	centos52	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 5.3	centos53	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 5.4	centos54	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 5.5	centos55	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 5.6	centos56	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 5.7	centos57	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 6.0	centos60	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 6.1	centos61	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 6.2	centos62	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 6.3	centos63	Section 8.5.1 [Installing with YUM], page 171
CentOS Enterprise Linux 6.4	centos64	Section 8.5.1 [Installing with YUM], page 171
Debian 4.0 Etch	deb4.0	Section 8.5.4 [Installing with APT], page 174
Debian 4.0r1 Etch	deb4.1	Section 8.5.4 [Installing with APT], page 174
Debian 4.0r2 Etch	deb4.2	Section 8.5.4 [Installing with APT], page 174
Debian 4.0r3 Etch	deb4.3	Section 8.5.4 [Installing with APT], page 174
Debian 5.0 Lenny	deb5.0	Section 8.5.4 [Installing with APT], page 174
Debian 6.0 Squeeze	deb6.0	Section 8.5.4 [Installing with APT], page 174
Debian 7.0 Wheezy	deb7.0	Section 8.5.4 [Installing with APT], page 174
Scientific Linux 5.0	sl5	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 5.1	sl51	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 5.2	sl52	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 5.3	sl53	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 5.4	sl54	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 5.5	sl55	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 5.6	sl56	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 5.7	sl57	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 6.0	sl6	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 6.1	sl61	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 6.2	sl61	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 6.3	sl61	Section 8.5.1 [Installing with YUM], page 171
Scientific Linux 6.4	sl61	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 5.0	puias5	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 5.1	puias51	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 5.2	puias52	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 5.3	puias53	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 5.4	puias54	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 5.5	puias55	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 5.6	puias56	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 5.7	puias57	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 6.0	puias6	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 6.1	puias61	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 6.2	puias61	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 6.3	puias61	Section 8.5.1 [Installing with YUM], page 171
PUIAS Linux 6.4	puias61	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 5.3	ols53	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 5.4	ols54	Section 8.5.1 [Installing with YUM], page 171

Oracle Linux Server 5.5	ols55	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 5.6	ols56	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 5.7	ols57	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 6.0	ols6	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 6.1	ols61	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 6.2	ols61	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 6.3	ols61	Section 8.5.1 [Installing with YUM], page 171
Oracle Linux Server 6.4	ols61	Section 8.5.1 [Installing with YUM], page 171
Fedora 7	FC7	Section 8.5.1 [Installing with YUM], page 171
Fedora 8	FC8	Section 8.5.1 [Installing with YUM], page 171
Fedora 9	FC9	Section 8.5.1 [Installing with YUM], page 171
Fedora 10	FC10	Section 8.5.1 [Installing with YUM], page 171
Fedora 11	FC11	Section 8.5.1 [Installing with YUM], page 171
Fedora 12	FC12	Section 8.5.1 [Installing with YUM], page 171
Fedora 13	FC13	Section 8.5.1 [Installing with YUM], page 171
Fedora 14	FC14	Section 8.5.1 [Installing with YUM], page 171
Fedora 15	FC15	Section 8.5.1 [Installing with YUM], page 171
Gentoo 2006.1	untested	Section 8.5.7 [Installing the Tar Ball], page 175
Gentoo 2007.1	untested	Section 8.5.7 [Installing the Tar Ball], page 175
RedHat Enterprise Linux 4	EL4	Section 8.5.1 [Installing with YUM], page 171
RedHat Enterprise Linux 5	EL5	Section 8.5.1 [Installing with YUM], page 171
RedHat Enterprise Linux 6	EL6	Section 8.5.1 [Installing with YUM], page 171
SuSE 10.0	SuSE10.0	Section 8.5.2 [Installing with ZYPPER], page 172
SuSE 10.1	SuSE10.1	Section 8.5.2 [Installing with ZYPPER], page 172
SuSE 10.2	SuSE10.2	Section 8.5.2 [Installing with ZYPPER], page 172
SuSE 10.3	SuSE10.3	Section 8.5.2 [Installing with ZYPPER], page 172
SuSE 11.0	SuSE11.0	Section 8.5.2 [Installing with ZYPPER], page 172
SuSE 11.1	SuSE11.1	Section 8.5.2 [Installing with ZYPPER], page 172
SuSE 11.2	SuSE11.2	Section 8.5.2 [Installing with ZYPPER], page 172
SuSE 11.3	SuSE11.3	Section 8.5.2 [Installing with ZYPPER], page 172
SuSE 11.4	SuSE11.4	Section 8.5.2 [Installing with ZYPPER], page 172
SLES 9 SP3	SLES9	Section 8.5.2 [Installing with ZYPPER], page 172
SLES 10	SLES10	Section 8.5.2 [Installing with ZYPPER], page 172
SLES 10 SP1	SLES10	Section 8.5.2 [Installing with ZYPPER], page 172
SLES 10 SP2	SLES10	Section 8.5.2 [Installing with ZYPPER], page 172
SLES 11	SLES11	Section 8.5.2 [Installing with ZYPPER], page 172
SLES 11 SP1	SLES11	Section 8.5.2 [Installing with ZYPPER], page 172
Mageia 1	MGA1	Section 8.5.3 [Installing with URPMI], page 173
Mandriva 2010.2	MDV2010.2	Section 8.5.3 [Installing with URPMI], page 173
Mandriva 2011.0	MDV2011.0	Section 8.5.3 [Installing with URPMI], page 173
MES	MDVMES5.2	Section 8.5.3 [Installing with URPMI], page 173
Ubuntu 8.10	ubu8.10	Section 8.5.4 [Installing with APT], page 174
Ubuntu 9.04	ubu9.04	Section 8.5.4 [Installing with APT], page 174
Ubuntu 9.10	ubu9.10	Section 8.5.4 [Installing with APT], page 174
Ubuntu 10.04	ubu10.04	Section 8.5.4 [Installing with APT], page 174
Ubuntu 10.10	ubu10.10	Section 8.5.4 [Installing with APT], page 174
Ubuntu 11.04	ubu11.04	Section 8.5.4 [Installing with APT], page 174
Ubuntu 11.10	ubu11.10	Section 8.5.4 [Installing with APT], page 174
Ubuntu 8.04 Server (LTS)	ubu8.04	Section 8.5.4 [Installing with APT], page 174

Ubuntu 10.04 Server (LTS)	ubu10.04	Section 8.5.4 [Installing with APT], page 174
Ubuntu 10.04.1 Server (LTS)	ubu10.04	Section 8.5.4 [Installing with APT], page 174
Ubuntu 10.04.2 Server (LTS)	ubu10.04	Section 8.5.4 [Installing with APT], page 174
Slackware 13.1	Slack13.1	⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩
Slackware 13.2	Slack13.2	⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩
Slackware 13.37	Slack13.37	⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩
Salix 13.1	Salix13.1	⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩
Salix 13.2	Salix13.2	⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩
Salix 13.37	Salix13.37	⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩
Archlinux	Archlinux	⟨undefined⟩ [⟨undefined⟩], page ⟨undefined⟩

8.1.2 Repositories for YUM

The `yum(8)` repositories are based on `repo-md` XML and are provided for all *RedHat*/*Fedora* architectures *RHEL 4.8* and beyond, or *FC 7* and beyond. This includes all current (supported) *RHEL* systems.

8.1.2.1 Setting up YUM

To install or upgrade from the *OpenSS7* `repo-md` repositories using `yum(8)`, you will need to install the `openss7-repo` RPM on your system, as follows:

```
$> REPO="https://user@www.openss7.org/repo/rpms"
$> REPONOARCH="$REPO/fedora/15/x86_64/RPMS/noarch"
$> REPORPM="$REPONOARCH/openss7-repo.noarch.rpm"
$> sudo rpm -Uhv $REPORPM
$> Username: anonymous
$> Password: *******
```

The example, above, assumes that the distribution is '`fedora`' and the distribution release is '`15`' and the required architecture is '`x86_64`'. Another example would be `$REPO/redhat/6.0/x86_64/RPMS/noarch`, for using `yum(8)` with RHEL.

To obtain access to the repository at the level to which you are entitled, you will have to respond to the '`Username:`' query with the user name with which you are registered with `https://www.openss7.org/`, and to the '`Password:`' query with your repository access password.[1]

You can test whether the `yum(8)` repository is properly set up by refreshing the repository and by listing the packages:

```
$> sudo yum makecache openss7
$> sudo yum search openss7
$> sudo yum info openss7
$> sudo yum info @openss7
```

Note that only the packages to which you are entitled will be listed.

[1] Note that your repository access password is separate and distinct from your web access or mailing list password.

8.1.2.2 Using the YUM Repository

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

```
$> sudo yum install openss7
$> sudo yum install @openss7
$> sudo yum groupinstall openss7
```

Removing the package is as easy as (one of):

```
$> sudo yum remove openss7
$> sudo yum remove @openss7
$> sudo yum groupremove openss7
```

To install the development packages for developing STREAMS modules and drivers, or applications that use the *OpenSS7 Protocol Suites*, install the development packages using (one of):

```
$> sudo yum install openss7-devel
$> sudo yum install @openss7-devel
$> sudo yum groupinstall openss7-devel
```

Of course, you are welcome to use a GUI based package manager, such as `PackageKit(8)`.

8.1.2.3 YUM Kernel Updates

The *OpenSS7* `yum(8)` repository definitions support the automatic updating of kernel modules when kernels are updated. When the system is updated, using '`yum update`', openss7 packages that are available for an updated kernel will be installed automatically. *OpenSS7* kernel module packages also support weak updates and when kernels are updated to compatible kernels (e.g. for security updates, or normally for any update within an Enterprise distribution), kernel modules are automatically applied against the updated kernel.

8.1.2.4 Setting up ZYPPER

To install or upgrade from the *OpenSS7* `repo-md` repositories using `zypper(8)`, you will need to install the `openss7-repo` RPM on your system, as follows:

```
$> REPO="https://user@www.openss7.org/repo/rpms"
$> REPONOARCH="$REPO/opensuse/11/x86_64/RPMS/noarch"
$> REPORPM="$REPONOARCH/openss7-repo.noarch.rpm"
$> sudo rpm -Uhv $REPORPM
$> Username: anonymous
$> Password: *******
```

The example assumes that the distribution is '`opensuse`' and the distribution release is '`11`' and the architecture requiresd is '`x86_64`'. Another example would be `$REPO/suse/11/x86_64/RPMS/noarch`, for using `zypper(8)` with SLES.

To obtain access to the repository at the level to which you are entitled, you will have to respond to the 'Username:' query with the user name with which you are registered with https://www.openss7.org/, and to the 'Password:' query with your repository access password.[2]

zypper(8) does not do a very good job of importing GPG signatures. When zypper is reporting that there are no keys for the RPMs being installed, the key can be imported directly into RPM as follows:

```
$> sudo rpm --import https://www.openss7.org/pubkey.asc
```

You can test whether the zypper(8) repository is properly set up by refreshing the repository and by listing the packages:

```
$> sudo zypper refresh openss7
$> sudo zypper search -t package openss7
$> sudo zypper info -t package openss7
```

8.1.2.5 Using the ZYPPER Repository

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

```
$> sudo zypper install -t pattern openss7
$> sudo zypper install openss7
```

Removing the package is as easy as:[3]

```
$> sudo zypper remove openss7
```

To install the development packages for developing *STREAMS* modules and drivers, or applications that use the *OpenSS7 Protocol Suites*, install the development packages using (one of):

```
$> sudo zypper install -t pattern openss7-devel
$> sudo zypper install openss7-devel
```

Of course, you are welcome to use a GUI based package manager, such as yast2(8).

8.1.2.6 ZYPPER Kernel Updates

The *OpenSS7* zypper(8) repository definitions support the automatic updating of kernel modules when kernels are updated. When the system is updated, using 'zypper update', *OpenSS7* packages that are available for an updated kernel will be installed automatically. *OpenSS7* kernel module packages also support weak updates and when kernels are updated to compatible kernels (e.g. for security updates, or normally for any update within an Enterprise distribution), kernel modules are automatically applied against the updated kernel.

[2] Note that your repository access password is separate from you web access or mailing list password.

[3] Note that zypper(8) does not yet support removal by pattern.

8.1.2.7 General REPO-MD Repository

To avoid having to change the `openss7.repo` file contents if they should change on the archive, place the following into the `openss7.repo` and place it into your `/etc/yum.repo.d/` directory:

```
-| include=https://www.openss7.org/repo/rpms/centos/$releasever/$arch/repodata/openss7.repo
```

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]
-| name = OpenSS7 Repository
-| baseurl = https://www.openss7.org/repo/rpms/centos/$releasever/$arch
-| gpgkey = https://www.openss7.org/repo/tarballs/OPENSS7-GPG-KEY
-| repo_gpgcheck = 1
-| gpgcheck = 1
-| enabled = 1
```

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.

The category that is provided is as follows:

'`openss7-components`' – OpenSS7 STREAMS and Protocol Suite Components.

The groups (patterns) that are provided in this category are as follows:

'`openss7`' — OpenSS7 STREAMS and Protocol Suites. This group (pattern) installs packages required for the *Linux Fast-STREAMS* and *OpenSS7 Protocol Suites* run-time libraries, commands, utilities, init scripts and kernel modules. Also included are section 1, 4, 5 and 8 manual pages for the commands, module and drivers, configuration file formats and administrative utilities.

Install this group (pattern) if you need the *Linux Fast-STREAMS* and *OpenSS7 Protocol Suties* core run-time.

The mandatory (required) packages in this group (pattern) are '`openss7`', '`openss7-base`', '`openss7-lib`' and '`openss7-kernel`'. The default (recommended) packages in this group (pattern) are '`openss7-java`'. The optional (suggested) packages in this group (pattern) are '`openss7-devel`', '`openss7-devel-kernel`', '`openss7-doc`', '`openss7-javadoc`' and '`openss7-source-kernel`'.

'`openss7-java`' — OpenSS7 STREAMS and Protocol Suite Java Components.
This group (pattern) installs packages required for the *Linux Fast-STREAMS* and *OpenSS7 JAIN* Java components.

The mandatory (required) packages in this group (pattern) are '`openss7-java`'. The optional (suggested) packages in this group (pattern) are '`openss7-javadoc`'.

'`openss7-devel`' — OpenSS7 STREAMS and Protocol Suite Development.
The mandatory (required) packages in this group (pattern) are '`openss7-devel`' and '`openss7-devel-kernel`'. The optional (suggested) packages in this group (pattern) are '`openss7-doc`', '`openss7-javadoc`' and '`openss7-source-kernel`'.

'`openss7-doc`' — OpenSS7 STREAMS and Protocol Suite Documentation.
The mandatory (required) packages in this group (pattern) are '`openss7-doc`' and '`openss7-javadoc`'.

For assistance with specific RPMs, see Section 8.2.5 [Downloading the Binary RPM], page 121.

8.1.3 Repositories for YAST

For distributions that support *YaST*, such as *SUSE Linux Enterprise Server*, and *OpenSUSE*, *YaST* repositories are built coexistent with *repo-md* repositories. Setting up the installation source as a *YaST* installation source instead of *repo-md* can provide the additional features associated with *YaST* install sources.

8.1.3.1 Setting up YAST

```
$> REPODIR="https://www.openss7.org/repo/rpms/suse/11/x86_64"
$> sudo zypper addrepo --type yast2 $REPODIR openss7
```

8.1.4 Repositories for URPMI

For distributions that support `urmpi(8)`, such as *Mageia*, *Mandriva* and *MES*, `urpmi(8)` repositories are built. Setting up the installation source as a `urpmi(8)` installation source instead of *repo-md* can provide the additional features associated with `urpmi(8)` install sources.

8.1.4.1 Setting up URPMI

To install or upgrade from the *OpenSS7* `urpmi` repositories using `urpmi(8)`, you will need to install the `openss7-repo` RPM on your system as follows:

```
$> REPO="https://user@www.openss7.org/repo/repms"
$> REPONOARCH="$REPO/mageia/1/x86_64/RPMS/noarch"
$> REPORPM="$REPONOARCH/openss7-repo.noarch.rpm"
$> sudo rpm -Uhv $REPORPM
$> Username: anonymous
$> Password: *******
```

The example, above, assumes that the distribution is '`mageia`' and the distribution release is '`1`' and the required architecture is '`x86_64`'. Another example would be `$REPO/mes/5.2/x86_64/RPMS/noarch`, for using `urpmi(8)` with MES (Mandriva Enterprise Server).

To obtain access to the respository at the level to which you are entitled, you will have to respond to the '`Username:`' query with the user name with which you are registered with `http://www.openss7.org/`, and to the '`Password:`' query with your repository access password.[4]

You can test whether the `urpmi(8)` repository is properly set up by refreshing the repository with `urpmi.update(8)` and by listing the packages with `urpmq(8)`:

```
$> sudo urpmi.update
$> sudo urpmq -a | grep openss7
$> sudo urpmq -ivl openss7
```

Note that only the packages to which you are entitled will be listed.

[4] Note that your repository access password is separate and distinct from your web access or mailing list password.

8.1.4.2 Using the URPMI Repository

Once the repository is set up, *OpenSS7* include a number of virtual package and package group definitions that ease the installation and removal of kernel modules, libriaries and utilities. Downloading, configuring, building and installation for a single-kernel distribution is as easy as (one of):

```
$> sudo urpmi openss7
```

Removing the package is as easy as (one of):

```
$> sudo urpme openss7
```

To install the development packages for developing STREAMS modules and drivers, or applications that use the *OpenSS7 Protocol Suites*, install the development packages using (one of):

```
$> sudo urpmi openss7-devel
```

Of course, you are welcome to use a GUI based package manager, such as `rpmdrake(8)`.

8.1.4.3 URPMI Kernel Updates

The *OpenSS7* `urpmi(8)` repository definitions support the automatic updating of kernel modules when kernels are updated. When the system is updated, using '`urpmi`', openss7 packages that are available for an updated kernel will be installed automatically. *OpenSS7* kernel module packages also support weak updates and when kernels are updated to compatible kernels (e.g. for security updates, or normally for any update within an Enterprise distribution), kernel modules are automatically applied against the updated kernel.

8.1.5 Repositories for APT

For assistance with specific DEBs, see Section 8.2.6 [Downloading the Debian DEB], page 125.

8.1.5.1 Setting up APT

To install or upgrade from the *OpenSS7* `apt` repositories using `apt(8)`, you will need to install the `openss7-repo` DEB on your system, as follows:

```
$> REPO="https://user@www.openss7.org/repo/debs"
$> REPONOARCH="$REPO/debian/squeeze/amd64"
$> REPODEB="$REPONOARCH/openss7-repo_all.deb"
$> wget $REPODEB
$> dpkg -i -D010077 openss7-repo_all.deb
$> Username: anonymous
$> Password: ********
```

The example, above, assumes that the distribution is '`debian`' and the distribution release is '`squeeze`' and the required architecture is '`amd64`'.[5] Another example would be `$REPO/ubuntu/10.04/amd64`, for using `apt(8)` with Ubuntu.

[5] Note that this is the '`$(uname -m)`' style architecture and not the Debian sytle architecture.

To obtain access to the repository at the level to which you are entitled, you will have to respond to the '`Username:`' query with the user name with which you are registered with `https://www.openss7.org/`, and to the '`Password:`' query with your repository access password.[6]

`apt(8)` does not do a very good job of importing GPG signatures. When `apt-get` is reporting that there are no keys for the DEBs being installed, the key can be imported directly into APT as follows:

```
$> wget https://www.openss7.org/pubkey.asc
$> sudo apt-key add pubkey.asc
```

You can test whether the `apt(8)` repository is properly set up by refreshing the repository and by listing the packages.

```
$> sudo apt-get update
$> aptitude search openss7
```

8.1.5.2 Using the APT Repository

Once the repository is set up, *OpenSS7* includes a number of virtual packages and packages recommendations that ease the installation and removal of kernel modules, libraries and utilities. Downloading, configuring, building and installation for a single-kernel distribution is as easy as:[7]

```
$> sudo aptitude install openss7
```

Removing the package is as easy as:

```
$> sudo aptitude remove openss7
$> sudo aptitude purge openss7
```

To install the development packages for developing *STREAMS* modules and drivers, or applications that use the *OpenSS7 Protocol Suites*, install the development packages using:

```
$> sudo aptitude install openss7-devel
```

Of course, you are welcome to use a GUI based package manager, such as `synaptic(8)`.

8.1.5.3 APT Kernel Updates

The *OpenSS7* `apt(8)` repository definitions support the automatic updating of kernel modules when kernels are updated. When the system is updated, using '`apt-get upgrade`', *OpenSS7* packages that are available for an updated kernel will be installed automatically.

OpenSS7 kernel module packages also support weak updates and when kernels are updated to compatible kernels (e.g. for security upgrades, or normally for any updated within an Enterprise (LTS) distribution), kernel modules are automatically applied against the updated kernel.

[6] Note that your repository access password is separate and distinct from your web access password.

[7] Note that using `aptitude(8)` in this way relies upon recommendations being treated as strong dependencies. This is normally the case for default `apt(8)` configuration files (particularly when `synaptic` is installed); however, it is not necessarily the case. If you are not getting recommended packages being automatically installed, add `--with-recommends` to the command line.

8.1.5.4 Setting up APT-RPM

To install or upgrade from the *OpenSS7* `apt-rpm` repositories using `apt(8)`, you will need to install the `openss7-repo` RPM on your system, as follows:

```
$> REPO="https://user@www.openss7.org/repo/rpms"
$> REPONOARCH="$REPO/mageia/1/x86_64/RPMS/noarch"
$> REPORPM="$REPONOARCH/openss7-repo.noarch.rpm"
$> wget $REPORPM
$> sudo rpm -Uhv $REPORPM
$> Username: anonymous
$> Password: *******
```

The example, above, assumes that the distribution is '`mageia`' and the distribution release is '`1`' and the required architecture is '`x86_64`'. Another example would be `$REPO/mes/5.2/x86_64/RPMS/noarch`, for using `apt(8)` with MES (Mandriva Enterprise Server).

To obtain access to the repository at the level to which you are entitled, you will have to respond to the '`Username:`' query with the user name with which you are registered with `https://www.openss7.org/`, and to the '`Password:`' query with your repository access password.[8]

`apt(8)` does not do a very good job of importing GPG signatures. When `apt-get` is reporting that there are no keys for the DEBs being installed, the key can be imported directly into APT as follows:

```
$> wget https://www.openss7.org/pubkey.asc
$> sudo apt-key add pubkey.asc
```

You can test whether the `apt(8)` repository is properly set up by refreshing the repository and by listing the packages.

```
$> sudo apt-get update
$> sudo apt-cache gencaches
$> apt-cache search openss7
```

8.1.5.5 Using the APT-RPM Repository

Once the repository is set up, *OpenSS7* includes a number of virtual packages and packages recommendations that ease the installation and removal of kernel modules, libraries and utilities. Downloading, configuring, building and installation for a single-kernel distribution is as easy as:[9]

```
$> sudo apt-get install openss7
```

Removing the package is as easy as:

[8] Note that your repository access password is separate and distinct from your web access password.

[9] Note that using `apt-get(8)` in this way relies upon recommendations being treated as strong dependencies. This is normally the case for default `apt(8)` configuration files (particularly when `synaptic` is installed); however, it is not necessarily the case. If you are not getting recommended packages being automatically installed, add `--install-recommends` to the command line.

```
$> sudo apt-get remove openss7
$> sudo apt-get purge openss7
```

To install the development packages for developing *STREAMS* modules and drivers, or applications that use the *OpenSS7 Protocol Suites*, install the development packages using:

```
$> sudo apt-get install openss7-devel
```

Of course, you are welcome to use a GUI based package manager, such as `synaptic(8)`.

8.1.5.6 APT-RPM Kernel Updates

The *OpenSS7* `apt(8)` repository definitions support the automatic updating of kernel modules when kernels are updated. When the system is updated, using '`apt-get upgrade`', *OpenSS7* packages that are available for an updated kernel will be installed automatically.

OpenSS7 kernel module packages also support weak updates and when kernels are updated to compatible kernels (e.g. for security upgrades, or normally for any updated within an Enterprise (LTS) distribution), kernel modules are automatically applied against the updated kernel.

8.1.6 Repositories for ALPM

8.1.6.1 Setting up ALPM

8.1.6.2 Using the ALPM Repository

8.1.6.3 ALPM Kernel Updates

8.2 Downloading

The *OpenSS7* 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 `OpenSS7` from the links in the sections that follow.

By far the easiest (most repeatable and manageable) form for installing and using the *OpenSS7* packages is to download and install the repository definition and use the distribution's native packaging tools. Another (still 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(8)` or `dpkg(1)`, the next best method for installing and using *OpenSS7* packages is to download and rebuild the source RPMs or DSCs.[10]

If your architecture does not support `rpm(8)` or `dpkg(1)` at all, or you have special needs (such as cross-compiling for embedded targets or for development), 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 is from tarball.[11]

[10] Note, however, that *OpenSS7 Corporation* restricts access to source RPMs and DSCs.
[11] Note, however, that *OpenSS7 Corporation* restricts access to tarballs.

8.2.1 Downloading with YUM

Once the repository definitions have been established, downloading RPMs with `yum(8)` is automatic. See Section 8.1.2.1 [Setting up YUM], page 112, for instructions on downloading the repository RPM and using it to set up for `yum(8)`, and Section 8.5.1 [Installing with YUM], page 171, for instructions on installing subsequent packages. See `yum(8)` for more information on downloading without installing.

8.2.2 Downloading with ZYPPER

Once the repository definitions have been established, downloading RPMs with `zypper(8)` is automatic. See Section 8.1.2.4 [Setting up ZYPPER], page 113, for instructions on downloading the repository RPM and using it to set up for `zypper(8)`, and Section 8.5.2 [Installing with ZYPPER], page 172, for instructions on installing subsequent packages. See `zypper(8)` for more information on downloading without installing.

8.2.3 Downloading with URPMI

Once the repository definitions have been established, downloading RPMs with `urpmi(8)` is automatic. See Section 8.1.4.1 [Setting up URPMI], page 116, for instructions on downloading the repository RPM and using it to set up for `urpmi(8)`, and Section 8.5.3 [Installing with URPMI], page 173, for instructions on installing subsequent packages. See `urpmi(8)` for more information on downloading without installing.

8.2.4 Downloading with APT

Once the repository definitions have been established, downloading DEBs with `apt(8)` is automatic. See Section 8.1.5.1 [Setting up APT], page 117, for instructions on downloading the repository DEB and using it to set up for `apt(8)`. See Section 8.1.5.4 [Setting up APT-RPM], page 119, for instructions on downloading the repository RPM and using it to set up for `apt(8)`. See Section 8.5.4 [Installing with APT], page 174, for instructions on installing subsequent packages. See `apt(8)` for more information on downloading without installing.

8.2.5 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 not dependent on the Linux kernel version. For example, the source package '`openss7-source-1.1.7.20141001-1.noarch.rpm`', is not dependent on kernel.

All of the following kernel 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 Section 8.4.1 [Building from the Source RPM], page 169).

Architecture Independent

Architecture independent RPMs are not dependent upon the processor architecture. That is, they are '`noarch`' RPMs. The architecture independent RPMs in the *OpenSS7* components are as follows:

`openss7-repo-1.1.7.20141001-1.noarch.rpm`
: This packages can be used to install or remove the repository source definitions for all OpenSS7 release packages. On systems that support repository access, this package is required by the other packages.

`openss7-1.1.7.20141001-1.noarch.rpm`
: This package can be used to install or remove the entire OpenSS7 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.

`openss7-base-1.1.7.20141001-1.noarch.rpm`
: The `openss7-base` binary package contains the init scripts, test scripts, maintenance scripts and base system configuration files necessary for the operation of Linux Fast-STREAMS and the protocol suites contained in the OpenSS7 package. It contains user and administrative documentation in `.info`, `.pdf` and `.html` formats as well as sections 1, 4, 5 and 8 of the manual pages. This binary package is required for any installation of the OpenSS7 package and is not architecture specific ('`noarch`').

`openss7-doc-1.1.7.20141001-1.noarch.rpm`
: The `openss7-doc` binary package contains the documentation used in the development of applications and programs that use the package. It contains developer and programmer manuals in `.info`, `.pdf` and `.html` formats as well as sections 2, 3, 7 and 9 of the manual pages and Javadoc HTML documentation. Install this binary package if you are interested in developing Linux Fast-STREAMS drivers or modules or application programs for the protocol suites contained in the OpenSS7 package. This package is massive and is not normally necessary except on a development system. The package is not architecture specific ('`noarch`').

`openss7-javadoc-1.1.7.20141001-1.noarch.rpm`
: The `openss7-javadoc` binary package contains Javadoc documentation for the OpenSS7 package. Install this binary package if you are interested in developing JAIN applications or resource adaptors. This package is not normally required on other than a development system. The package is not architecture specific ('`noarch`').

Architecture Dependent

Architecture dependent RPMs are dependent upon the processor architecture (but not on the specific kernel version). That is, they are '`i686`' RPMs. The architecture dependent RPMs in the *OpenSS7* components are as follows:

`openss7-lib-1.1.7.20141001-1.i686.rpm`
: The `openss7-lib` binary package contains the run-time (shared object) libraries necessary to run applications programs and utilities developed for *Linux Fast-STREAMS* and the protocol suites contained in the *OpenSS7* package. Also included are the libtool `.la` files describing the shared object libraries. The binary package also provides administrative and configuration test utilities and commands associated with the OpenSS7 package. Note that these utilities are needed for running the validation test suites contained in the `openss7-base` binary package.

`openss7-java-1.1.7.20141001-1.i686.rpm`
: The `openss7-java` binary package contains JAIN implementations for OpenSS7 Linux Fast-STREAMS and the associated protocol suites. It includes jar files, JNI and CNI libraries, and gcj native compiled libraries. It also includes SWIG implementations of Java interfaces for OpenSS7 API libraries.

`openss7-devel-1.1.7.20141001-1.i686.rpm`
: The `openss7-devel` binary package contains library archives for static compilation, and header files to develop Linux Fast-STREAMS modules and drivers. This also includes header files and static libraries required to compile Linux Fast-STREAMS and OpenSS7 protocol suite applications programs.

  This binary package does not contain developer nor programmer documentation nor manual pages. To obtain the developer and programmer documentation, load the `openss7-doc` binary package. This package is architecture-specific but not kernel-specific.

  This package can be used to install or remove the development components of the OpenSS7 package. When installing, '`openss7`' an appropriate kernel module and kernel module development packages will also be installed. When removing, the development package and all kernel module development packages will also be removed.

`openss7-debuginfo-1.1.7.20141001-1.i686.rpm`
: The `openss7-debuginfo` binary package contains debugging information stripped from C-language libraries and C-language binary executables. This package is required for full debugging of the executables contained in the `openss7-lib` binary package. This package provides debug information for the OpenSS7 packages. Debug information is useful when developing applications that use the OpenSS7 package or when debugging the OpenSS7 package.

  This package can be used to install or remove the debug information components of the OpenSS7 package. When installing, '`openss7`' an appropriate kernel module and kernel module debug information packages will also be installed. When removing, the debug information package and all kernel module debug information packages will also be removed.

`openss7-debugsource-1.1.7.20141001-1.i686.rpm`
: The `openss7-debugsource` binary package contains debugging source references stripped from C-language libraries and C-language binary executables. This package is required for source-level debugging of the executables contained in the `openss7-lib` binary package. This package provides debug source for the OpenSS7 packages. Debug source is useful when developing applications that use the OpenSS7 package or when debugging the OpenSS7 package.

  This package can be used to install or remove the debug source components of the OpenSS7 package. When installing, '`openss7`' an appropriate kernel module and kernel module debug source packages will also be installed. When removing, the debug source package and all kernel module debug source packages will also be removed.

8.2.5.1 Kernel-Dependent RPM

Kernel-Dependent RPM are dependent on specific Linux Kernel Binary RPM releases. Packages are provided for popular distribution production kernels. Packages dependent upon a 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 available, you can build binary RPM from the source RPM (see see Section 8.4.1 [Building from the Source RPM], page 169).[12]

`openss7-3.0.99-1-unx-1.1.7.20141001-1.i686.rpm`

The `openss7-kernel` binary package contains the kernel modules that provide the Linux kernel Linux Fast-STREAMS drivers and modules. This includes assorted drivers and modules for Linux Fast-STREAMS and additional OpenSS7 protocol suite components. This binary package also includes modprobe configuration files for the associated kernel modules. This package is heavily tied to the kernel for which it and dependent components were compiled. This package applies to kernel version 3.0.99-1-unx and requires components compiled for the same kernel. If you cannot find a binary package that matches your kernel, rebuild for your kernel from the OpenSS7 package source rpm.

This package can be used to install or remove the package for a specific kernel version. When installing, the '`openss7`' package will also be installed if necessary. When removing the last kernel module package, the '`openss7`' package will also be removed.

Note that the version '`3.0.99-1-unx`' is just an example. Use the version returned by '`$(uname -r)`' for the kernel for which you wish to install or remove the packages.

`openss7-devel-3.0.99-1-unx-1.1.7.20141001-1.i686.rpm`

The `openss7-devel-3.0.99-1-unx` binary package contains the kernel modules symbol information for development of additional Linux Fast-STREAMS kernel modules against the core kernel modules included in the '`openss7-3.0.99-1-unx`' binary package. This package is heavily tied to the core kernel modules and kernel for which it was compiled. This package applies to core kernel modules '`3.0.99-1-unx`' for kernel version '`3.0.99-1-unx`'. This package provides kernel debug information for the OpenSS7 package. Kernel debug information is useful when developing kernel modules that use this package or when debugging kernel modules contained in the package.

This package can be used to install or remove the development and debug packages for a specific kernel version. When installing, the '`openss7`' and '`openss7-devel`' packages will also be installed if necessary. When removing the development and debug for kernel modules for the last kernel, the '`openss7-devel`' package will also be removed.

Note that the version '`3.0.99-1-unx`' is just an example. Use the version returned by '`$(uname -r)`' for the kernel for which you wish to install or remove the packages.

The `openss7-devel-3.0.99-1-unx` package contains the module symbol version information for the `kernel` package, above. It is possible to load this package and compile modules that use the exported symbols without loading the actual kernel modules (from the `kernel` package above). This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version '`3.0.99-1-unx`'.[13]

`openss7-debuginfo-3.0.99-1-unx-1.1.7.20141001-1.i686.rpm`
`openss7-debugsource-3.0.99-1-unx-1.1.7.20141001-1.i686.rpm`
`openss7-source-3.0.99-1-unx-1.1.7.20141001-1.i686.rpm`

The `openss7-source-3.0.99-1-unx` package contains the configured source for a specific kernel. This package contains source that is configured to the kernel for which it

[12] 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.

[13] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example.

was configured. This particular package contains configured source for kernel version '`3.0.99-1-unx`'[14] The `openss7-source` package contains the source code necessary for building the OpenSS7 release. It includes the `autoconf(1)` configuration utilities necessary to create and distribute tarballs, `rpm` and `deb/dsc`.

The `openss7-source-3.0.99-1-unx` binary package contains the source code necessary for building the OpenSS7 release for the kernel version '`3.0.99-1-unx`'. It also includes the `autoconf(1)` configuration utilities necessary to create and distribute this rpm. However, to develop on the package, it would be better to use the tarball release. Load this package if you need some files from the build that are not present in the `openss7-devel-3.0.99-1-unx` package.

Configuration and Installation

To configure, build and install the binary RPM, see Section 8.3.1 [Configuring the Binary RPM], page 130.

8.2.6 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 not dependent on the Linux kernel version. For example, the source package '`openss7-source_1.1.7.20141001-0_i386.deb`', is not dependent on kernel.

All of the following kernel 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 8.4.2 [Building from the Debian DSC], page 170).

Architecture Independent

openss7-repo_1.1.7.20141001-0_all.deb
openss7-dev_1.1.7.20141001-0_all.deb
: The `openss7-dev` package contains the device definitions necessary to run applications programs developed for OpenSS7.[15]

openss7-doc_1.1.7.20141001-0_all.deb
: The `openss7-doc` package contains this manual in plain text, postscript, `pdf` and `html` forms, along with the meta-information from the `OpenSS7` package. It also contains all of the manual pages necessary for developing OpenSS7 applications and OpenSS7 STREAMS modules or drivers.

[14] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example.

[15] 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.

openss7-init_1.1.7.20141001-0_all.deb
: The `openss7-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.

openss7-source_1.1.7.20141001-0_all.deb
: The `openss7-source` package contains the source code necessary for building the OpenSS7 release. It includes the `autoconf(1)` configuration utilities necessary to create and distribute tarballs, rpms and deb/dscs.

Architecture Dependent

openss7-devel_1.1.7.20141001-0_i386.deb
: The `openss7-devel` package contains library archives for static compilation, header files to develop OpenSS7 modules and drivers. This also includes the header files and static libraries required to compile OpenSS7 applications programs.

openss7-lib_1.1.7.20141001-0_i386.deb
: The `openss7-lib` package contains the run-time shared libraries necessary to run application programs and utilities developed for the `OpenSS7` package.

openss7-debuginfo_1.1.7.20141001-0_i386.deb
openss7-debugsource_1.1.7.20141001-0_i386.deb

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 8.4.2 [Building from the Debian DSC], page 170).[16]

openss7-core-3.0.99-1-unx_1.1.7.20141001-0_i386.deb
: The `openss7-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 '`3.0.99-1-unx`'.[17]

openss7-info-3.0.99-1-unx_1.1.7.20141001-0_i386.deb
: The `openss7-info` package[18] 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 '`3.0.99-1-unx`'.[19]

[16] 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.

[17] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example. Note also that only release packages that contain kernel modules will contain a `core` subpackage.

[18] 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 or 3.x series kernels.

[19] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example.

Configuration and Installation

To configure, build and install the Debian DEB, see Section 8.3.2 [Configuring the Debian DEB], page 132.

8.2.7 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.

openss7-1.1.7.20141001-1.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, 2.6 or 3.x kernel.

Configuration

To configure the source RPM, see Section 8.3.3 [Configuring the Source RPM], page 132.

8.2.8 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.

openss7_1.1.7.20141001-0.dsc
openss7_1.1.7.20141001-0.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, 2.6 or 3.x kernel.

Configuration

To configure the source RPM, see Section 8.3.4 [Configuring the Debian DSC], page 143.

8.2.9 Downloading the Tar Ball

For non-`rpm(8)` and non-`dpkg(1)` architectures, download the tarball as follows:

openss7-1.1.7.20141001.tar.bz2
openss7-1.1.7.20141001.tar.xz
: 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(8)` architectures and Debian DSC and DEB on `dpkg(1)` architectures.

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

```
% wget https://www.openss7.org/repo/tarballs/openss7-1.1.7.20141001.tar.bz2
```

or

```
% wget https://www.openss7.org/repo/tarballs/openss7-1.1.7.20141001.tar.xz
```

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 https://www.openss7.org/repo/tarballs/openss7-1.1.7.20141001.tar.bz2
% tar -xjvf openss7-1.1.7.20141001.tar.bz2
```

or

```
% wget https://www.openss7.org/repo/tarballs/openss7-1.1.7.20141001.tar.xz
% tar -xJvf openss7-1.1.7.20141001.tar.xz
```

Either will create a subdirectory name `openss7-1.1.7.20141001` containing all of the files and subdirectories for the `OpenSS7` package.

Configuration

To configure and install the tar ball, see Section 8.3.5 [Configuring the Tar Ball], page 143.

8.2.10 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 `OpenSS7` package from the project CVS archive.

The OpenSS7 package is located in the `openss7` module of `/var/cvs`. For release tag information, see Chapter 7 [Releases], page 61.

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

```
% export CVSROOT=':pserver:username@cvs.openss7.org:2401/var/cvs'
% cvs login
Password: *********
% cvs co -r openss7_1.1.7.20141001 openss7
% 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 `OpenSS7` 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 1.4.17`
- `autoconf 2.69`
- `automake 1.14.1`
- `libtool 2.4.2`
- `gettext 0.19.2`
- `flex 2.5.39`
- `bison 3.0.2`
- `swig 3.0.2`
- `xz-5.0.7`

Most desktop development GNU/Linux distributions will have these tools; however, some non-development or server-style installations might not and they must be installed separately.[20]
Also, these tools can be acquired from the FSF website in the free software directory, and also at the following locations:

- m4-1.4.17
- autoconf-2.69
- automake-1.14.1
- libtool-2.4.2
- gettext-0.19.2
- flex-2.5.39
- bison-3.0.2
- swig-3.0.2
- xz-5.0.7

It should be stressed that, in particular, the `autoconf(1)`, and `automake(1)`, must be at version releases `2.68` and `1.11.1`. *The versions normally distributed in some mainstream GNU/Linux distributions are, in fact, much older than these versions.*[21] 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 9.15` or `ghostscript 9.15`, or newer.
- `tetex 3.14159265` or `texlive 2014`, or newer.
- `texinfo 5.2` or newer.
- `transfig 3.2.5e` or newer.
- `imagemagick 6.8.0.8` or `ImageMagick 6.8.0.8`, or newer.
- `groff 1.22.2` or newer.
- `gnuplot 4.6` or newer.
- `latex2html 2012 (1.2)` 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.[22]
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:

[20] 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 3.0.2 and flex 2.5.39.
[21] A notable exception is Debian and Fedora 7. Note that on Fedora 7 the `gettext-devel` package must be installed.
[22] 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.

```
% 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.

If you wish to package rpms on an rpm(8) system, or debs on a dpkg(1) system, you will need the appropriate tool chain. Systems based on rpm(8) 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 openss7
```

where, openss7 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 '2.68'. Otherwise, you may need to perform something like the following:

```
% PATH="/usr/local/bin:$PATH"
% autoreconf -fiv openss7
```

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 8.3.5 [Configuring the Tar Ball], page 143, and Section 8.4.3 [Building from the Tar Ball], page 170.

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 8.3.5 [Configuring the Tar Ball], page 143.

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.

8.3 Configuration

8.3.1 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:

`openss7-core-3.0.99-1-unx-1.1.7.20141001-1.i686.rpm`

`/lib/modules/3.0.99-1-unx`
: This relocatable directory contains the kernel modules that provide the OpenSS7 core, drivers and modules.[23]

`openss7-info-3.0.99-1-unx-1.1.7.20141001-1.i686.rpm`

`/usr/include/openss7/3.0.99-1-unx`
: This relocatable directory contains the kernel module exported symbol information that allows other kernel modules to be compiled against the correct version of the openss7 package.[24]

`openss7-dev-1.1.7.20141001-1.i686.rpm`
: (not relocatable)

`openss7-devel-1.1.7.20141001-1.i686.rpm`

`/usr/lib` This relocatable directory contains openss7 libraries.

`/usr/include/openss7`
: This relocatable directory contains openss7 header files.

`openss7-doc-1.1.7.20141001-1.i686.rpm`

`/usr/share/doc`
: This relocatable directory contains all package specific documentation (including this manual). The subdirectory in this directory is the `openss7-1.1.7.20141001` 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.

`openss7-lib-1.1.7.20141001-1.i686.rpm`

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

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

`openss7-source-1.1.7.20141001-1.i686.rpm`

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

`openss7-util-1.1.7.20141001-1.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.

[23] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example.

[24] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example. Also, note that the '`info`' subpackage is only applicable to the 2.4 kernel series.

Installation

To install the binary RPM, see Section 8.5.5 [Installing the Binary RPM], page 175.

8.3.2 Configuring the Debian DEB

In general the binary DEB do not require any configuration.

Installation

To install the Debian DEB, see Section 8.5.6 [Installing the Debian DEB], page 175.

8.3.3 Configuring the Source RPM

When building from the source RPM (see Section 8.4.1 [Building from the Source RPM], page 169), 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 8.4.1 [Building from the Source RPM], page 169), 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.

`--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.[25] 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. 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

[25] 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.

from functions and invoking most debugging macros in the code. The default is to not perform kernel testing.

`--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 8.3.5 [Configuring the Tar Ball], page 143.

`--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.

`--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.

`--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 run-time 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 run-time 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.

In addition, the following `rpm` options, specific to the OpenSS7 package are available:

`--without streams-irq`
: Disables STREAMS irq suppression. Normally the STREAMS scheduler protects itself and its datastructures from races due to hard interrupts by suppressing interrupts during critical sections. As not all drivers and modules contain hard interrupts, this option allows hard interrupts to be enabled while running critical sections. The purpose of this option was primarily for profiling. The default is to enable STREAMS irq suppression.

`--without streams-stats`
: Disable STREAMS statistics counting. Normally the STREAMS scheduler will automatically count the number of entries to open, close, put and service procedures for a queue pair whenever the `module_stats` structure is defined and attached to the `qinit` structure by the module or driver. This is not exact SVR4.2 MP behaviour (where it is the responsibility of the module or drive to perform these counts). This option disables the feature. The default is to enable STREAMS statistics counting.

`--without streams-syncqs`
: Disable STREAMS synchronization queues. Normally the STREAMS scheduler will permit modules and drivers that are written for syncrhonization (such as SVR4.2 MP synchronization, Solaris perimeters, etc.) and will perform synchronization protection for these modules. This option disables synchronization queues. When disabled, only fully multiprocessor safe drivers and modules (marked with the D_MP flag), will be loaded. The default is to enable STREAMS synchronization queues.

`--without streams-utils`
: Disable additional STREAMS utilities. Normally `strsetup` and `strload` utility configuration files are included in the build and installed. This option disables build and installation of the `strsetup` and `strload` configuration files. The default is to enable additional STREAMS utilities.

`--without big-compile`
: Disable compilation as one big computational unit. The default is to build as one big computational unit. Do not use this option.

`--with streams-fifos`
: Enable overriding of system FIFOs with STREAMS-based FIFOs. The default is to not override system FIFOs with STREAMS-based FIFOs.

`--with streams-kthreads`
: Set STREAMS kernel thread operation to *nice*, *normal*, *rt* or *no*. When set to *nice*, the STREAMS scheduler will be based on kernel threads that run with a '`nice -19`' priority. When set to *normal*, the STREAMS scheduler sill be based on kernel threas thar run with '`nice -0`' priority. When set to *rt*, the STREAMS scheduler will be based on kernel threads that run with real-time priority '`nice -99`'. When set to *no*, soft-interrupts will be used for the STREAMS scheduler rather than kernel threads.

This option was primarily for performance testing. Do no use this option. The default STREAMS kernel thread pirority is *nice*.

`--with module-sth`
`--without module-sth`
: Enables or disables the `sth` (Stream head) module. When enabled, the `sth` module will be compiled into the `streams` kernel module; when disabled, the `sth` module will not be included at all. Note that disabling the `sth` module will cause all Streams to fail. The default is for the `sth` module to be created as a separate module.

`--with module-srvmod`
`--without module-srvmod`
: Enables or disables the `srvmod` module. When enabled, the `srvmod` module will be compiled into the `streams` kernel module; when disabled, the `srvmod` module will not be included at all. Note that disabling the `srvmod` module will cause conformance suites to fail. The default is for the `srvmod` module to be created as a separate module.

`--with module-nullmod`
`--without module-nullmod`
: Enables or disables the `nullmod` module. When enabled, the `nullmod` module will be compiled into the `streams` kernel module; when disabled, the `nullmod` module will not be included at all. Note that disabling the `nullmod` module will cause conformance suites to fail. The default is for the `nuls` module to be created as a separate module.

`--with module-pipemod`
`--without module-pipemod`
: Enables or disables the `pipemod` module. When enabled, the `pipemod` module will be compiled into the `streams` kernel module; when disabled, the `pipemod` module will not be included at all. Note that disabling the `pipemod` module will cause conformance suites to fail. The default is for the `pipemod` module to be created as a separate module.

`--with module-connld`
`--without module-connld`
: Enables or disables the `connld` module. When enabled, the `connld` module will be compiled into the `streams` kernel module; when disabled, the `connld` module will not be included at all. Note that disabling the `connld` module will cause conformance suites to fail. The default is for the `connld` module to be created as a separate module.

`--with module-sc`
`--without module-sc`
: Enables or disables the `sc` module. When enabled, the `sc` module will be compiled into the `streams` kernel module; when disabled, the `sc` module will not be included at all. Note that disabling the `sc` module will cause conformance suites to fail. The default is for the `sc` module to be created as a separate module.

`--with module-testmod`
`--without module-testmod`
: Enables or disables the `testmod` module. When enabled, the `testmod` module will be compiled into the `streams` kernel module; when disabled, the `testmod` module will not be included at all. Note that disabling the `testmod` module will cause conformance suites to fail. The default is for the `testmod` module to be created as a separate module.

`--with module-timod`
`--without module-timod`
: Enables or disables the `timod` module. When enabled, the `timod` module will be compiled into the `streams` kernel module; when disabled, the `timod` module will not be included at all. Note that disabling the `timod` module will cause conformance suites to fail. The default is for the `timod` module to be created as a separate module.

`--with module-tirdwr`
`--without module-tirdwr`
: Enables or disables the `tirdwr` module. When enabled, the `tirdwr` module will be compiled into the `streams` kernel module; when disabled, the `tirdwr` module will not be included at all. Note that disabling the `tirdwr` module will cause conformance suites to fail. The default is for the `tirdwr` module to be created as a separate module.

`--with module-bufmod`
`--without module-bufmod`
: Enables or disables the `bufmod` module. When enabled, the `bufmod` module will be compiled into the `streams` kernel module; when disabled, the `bufmod` module will not be included at all. Note that disabling the `bufmod` module will cause conformance suites to fail. The default is for the `bufmod` module to be created as a separate module.

`--with module-pfmod`
`--without module-pfmod`
: Enables or disables the `pfmod` module. When enabled, the `pfmod` module will be compiled into the `streams` kernel module; when disabled, the `pfmod` module will not be included at all. Note that disabling the `pfmod` module will cause conformance suites to fail. The default is for the `pfmod` module to be created as a separate module.

`--with module-nbuf`
`--without module-nbuf`
: Enables or disables the `nbuf` module. When enabled, the `nbuf` module will be compiled into the `streams` kernel module; when disabled, the `nbuf` module will not be included at all. Note that disabling the `nbuf` module will cause conformance suites to fail. The default is for the `nbuf` module to be created as a separate module.

`--with module-pf`
`--without module-pf`
: Enables or disables the `pf` module. When enabled, the `pf` module will be compiled into the `streams` kernel module; when disabled, the `pf` module will not be included at all. Note that disabling the `pf` module will cause conformance suites to fail. The default is for the `pf` module to be created as a separate module.

`--with driver-clone`
`--without driver-clone`
: Enables or disables the `clone` module. When enabled, the `clone` module will be compiled into the `streams` kernel module; when disabled, the `clone` module will not be included at all. Note that disabling the `clone` module will cause conformance suites to fail. The default is for the `clone` driver to be created as a separate module.

`--with driver-echo`
`--without driver-echo`
: Enables or disables the `echo` module. When enabled, the `echo` module will be compiled into the `streams` kernel module; when disabled, the `echo` module will not be included

at all. Note that disabling the `echo` module will cause conformance suites to fail. The default is for the `echo` driver to be created as a separate module.

`--with driver-fifo`
`--without driver-fifo`
: Enables or disables the `fifo` module. When enabled, the `fifo` module will be compiled into the `streams` kernel module; when disabled, the `fifo` module will not be included at all. Note that disabling the `fifo` module will cause conformance suites to fail. The default is for the `fifo` driver to be created as a separate module.

`--with driver-log`
`--without driver-log`
: Enables or disables the `log` module. When enabled, the `log` module will be compiled into the `streams` kernel module; when disabled, the `log` module will not be included at all. Note that disabling the `log` module will cause conformance suites to fail. The default is for the `log` driver to be created as a separate module.

`--with driver-loop`
`--without driver-loop`
: Enables or disables the `loop` module. When enabled, the `loop` module will be compiled into the `streams` kernel module; when disabled, the `loop` module will not be included at all. Note that disabling the `loop` module will cause conformance suites to fail. The default is for the `loop` driver to be created as a separate module.

`--with driver-nsdev`
`--without driver-nsdev`
: Enables or disables the `nsdev` module. When enabled, the `nsdev` module will be compiled into the `streams` kernel module; when disabled, the `nsdev` module will not be included at all. Note that disabling the `nsdev` module will cause conformance suites to fail. The default is for the `nsdev` driver to be created as a separate module.

`--with driver-mux`
`--without driver-mux`
: Enables or disables the `mux` module. When enabled, the `mux` module will be compiled into the `streams` kernel module; when disabled, the `mux` module will not be included at all. Note that disabling the `mux` module will cause conformance suites to fail. The default is for the `mux` driver to be created as a separate module.

`--with driver-nuls`
`--without driver-nuls`
: Enables or disables the `nuls` module. When enabled, the `nuls` module will be compiled into the `streams` kernel module; when disabled, the `nuls` module will not be included at all. Note that disabling the `nuls` module will cause conformance suites to fail. The default is for the `nuls` driver to be created as a separate module.

`--with driver-pipe`
`--without driver-pipe`
: Enables or disables the `pipe` module. When enabled, the `pipe` module will be compiled into the `streams` kernel module; when disabled, the `pipe` module will not be included at all. Note that disabling the `pipe` module will cause conformance suites to fail. The default is for the `pipe` driver to be created as a separate module.

`--with driver-sad`
`--without driver-sad`
: Enables or disables the `sad` module. When enabled, the `sad` module will be compiled into the `streams` kernel module; when disabled, the `sad` module will not be included at all. Note that disabling the `sad` module will cause conformance suites to fail. The default is for the `sad` driver to be created as a separate module.

`--with driver-sfx`
`--without driver-sfx`
: Enables or disables the `sfx` module. When enabled, the `sfx` module will be compiled into the `streams` kernel module; when disabled, the `sfx` module will not be included at all. Note that disabling the `sfx` module will cause conformance suites to fail. The default is for the `sfx` driver to be created as a separate module.

`--with driver-spx`
`--without driver-spx`
: Enables or disables the `spx` module. When enabled, the `spx` module will be compiled into the `streams` kernel module; when disabled, the `spx` module will not be included at all. Note that disabling the `spx` module will cause conformance suites to fail. The default is for the `spx` driver to be created as a separate module.

`--with compat-os7`
`--without compat-os7`
: Enables or disables the `os7` compatibility module. When enabled, the `os7` compatibility module will be compiled into the `stream` kernel module; when disabled, the `os7` compatibility module will not be included at all. The default is for the `os7` compatibility module to be created as a separate module.

`--with compat-svr3`
`--without compat-svr3`
: Enables or disables the `svr3` compatibility module. When enabled, the `svr3` compatibility module will be compiled into the `stream` kernel module; when disabled, the `svr3` compatibility module will not be included at all. The default is for the `svr3` compatibility module to be created as a separate module.

`--with compat-svr4`
`--without compat-svr4`
: Enables or disables the `svr4` compatibility module. When enabled, the `svr4` compatibility module will be compiled into the `stream` kernel module; when disabled, the `svr4` compatibility module will not be included at all. The default is for the `svr4` compatibility module to be created as a separate module.

`--with compat-mps`
`--without compat-mps`
: Enables or disables the `mps` compatibility module. When enabled, the `mps` compatibility module will be compiled into the `stream` kernel module; when disabled, the `mps` compatibility module will not be included at all. The default is for the `mps` compatibility module to be created as a separate module.

`--with compat-sol8`
`--without -without l8`
: Enables or disables the `sol8` compatibility module. When enabled, the `sol8` compatibility module will be compiled into the `stream` kernel module; when disabled, the

`sol8` compatibility module will not be included at all. The default is for the `sol8` compatibility module to be created as a separate module.

`--with compat-uw7`
`--without compat-uw7`
: Enables or disables the `uw7` compatibility module. When enabled, the `uw7` compatibility module will be compiled into the `stream` kernel module; when disabled, the `uw7` compatibility module will not be included at all. The default is for the `uw7` compatibility module to be created as a separate module.

`--with compat-osf`
`--without compat-osf`
: Enables or disables the `osf` compatibility module. When enabled, the `osf` compatibility module will be compiled into the `stream` kernel module; when disabled, the `osf` compatibility module will not be included at all. The default is for the `osf` compatibility module to be created as a separate module.

`--with compat-aix`
`--without compat-aix`
: Enables or disables the `aix` compatibility module. When enabled, the `aix` compatibility module will be compiled into the `stream` kernel module; when disabled, the `aix` compatibility module will not be included at all. The default is for the `aix` compatibility module to be created as a separate module.

`--with compat-hpux`
`--without compat-hpux`
: Enables or disables the `hpux` compatibility module. When enabled, the `hpux` compatibility module will be compiled into the `stream` kernel module; when disabled, the `hpux` compatibility module will not be included at all. The default is for the `hpux` compatibility module to be created as a separate module.

`--with compat-irix`
`--without compat-irix`
: Enables or disables the `irix` compatibility module. When enabled, the `irix` compatibility module will be compiled into the `stream` kernel module; when disabled, the `irix` compatibility module will not be included at all. The default is for the `irix` compatibility module to be created as a separate module.

`--with compat-mac`
`--without compat-mac`
: Enables or disables the `mac` compatibility module. When enabled, the `mac` compatibility module will be compiled into the `stream` kernel module; when disabled, the `mac` compatibility module will not be included at all. The default is for the `mac` compatibility module to be created as a separate module.

`--without xti-servtype`
: Disables XTI service type checks in the XTI/TLI Library. Normally, the XTI/TLI Library will check for the service type of the endpoint and will reject commands that are not defined for the corresponding service type. When enabled, this option causes the XTI/TLI Library to simply issue the corresponding primitive to the underlying driver and to allow the driver to determine whether the primitive is supported. The default is for the XTI/TLI Library to check for XTI service type.

`--without xti-states`
: Disables XTI state checks in the XTI/TLI Library. Normally the XTI/TLI Library will check for the state of the endpoint and will reject commands that would place the interface out of state. When enabled, this option causes the XTI/TLI Library to simply issue the corresponding primitive to the underlying driver and to allow the driver to determine whether the interface is out of state. The default is for the XTI/TLI Library to check for XTI state.

`--with sctp-slow-verification`
: Enable slow verification of addresses and tags. When a message comes from an SCTP endpoint with the correct verification tag, it is not necessary to check whether it is from a correct source address to identify the SCTP association to which it belongs. When you disable this feature (`--without sctp-slow-verification`), source addresses are not checked and it is up to firewall implementations to thwart attackers of the verification tag. When you enable this feature (`--with sctp-slow-verification`), you get RFC 2960 compliant operation, but at great cost to SCTP performance. This option defaults to '`disabled`'.

`--with sctp-throttle-heartbeats`
: Enable heartbeat throttling. Special feature of OpenSS7 that is not mentioned in RFC 2960. When you enable this feature (`--with sctp-throttle-heartbeats`), OpenSS7 will throttle the rate at which it responds to heartbeats to the system control *heartbeat_interval*. This makes SCTP more resilient to implementations which flood heartbeat messages. For RFC 2960 compliant operation, disable this feature (`--without sctp-throttle-heartbeats`). This option defaults to '`disabled`'.

`--with sctp-discard-ootb`
: Enable discard of out-of-the-blue packets. RFC 2960 requires the implementation to send `ABORT` to some OOTB packets (packets for which no SCTP association exists). Sending `ABORT` chunks to unverified source addresses with the T bit set opens SCTP to blind masquerade attacks. Not sending them may lead to delays at the peer endpoint aborting associations where our `ABORT` has been lost and the socket is already closed or if we have restarted and the peer still has open associations to us. If you enable this feature (`--with sctp-discard-ootb`), SCTP will discard all OOTB packets. This is necessary if another SCTP stack is being run on the same machine. Therefore, if the OpenSS7 package is included on an OpenSS7 SCTP kernel, this feature is automatically enabled. For RFC 2960 compliant operation, disable this feature (`--without sctp-discard-ootb`). This option defaults to '`disabled`' for non-OpenSS7 SCTP kernels, and '`enabled`' for OpenSS7 SCTP kernels.

`--with sctp-exteded-ip-support`
: Enable extended IP support for SCTP. This provides extended IP support for SCTP for things like IP Transparent Proxy and IP Masquerading. This is experimental stuff. If in doubt, disable this feature (`--without sctp-expended-ip-support`). This option defaults to '`disabled`'.

`--without sctp-hmac-sha1`
: Disable SHA-1 HMAC. This provides the ability to use the FIPS 180-1 (SHA-1) message authentication code in SCTP cookies. If you enable this feature (`--with sctp-hmac-sha1`), when the appropriate sysctl is set, SCTP will use the SHA-1 HMAC when signing cookies in the `INIT-ACK` chunk. If disable this feature (`--without sctp-hmac-`

`sha1`), the SHA-1 HMAC will be unavailable for use with SCTP. This option defaults to '`enabled`' on big-endian architectures, and '`disabled`' otherwise.

`--without sctp-hmac-md5`
: Disable MD5 HMAC. This provides the ability to use the MD5 (RFC 1321) message authentication code in SCTP cookies. If you enable this feature (`--with sctp-hmac-md5`), when the appropriate sysctl is set, SCTP will use the MD5 HMAC when signing cookies in the INIT ACK chunk. If you disable this feature (`--without sctp-hmac-md5`), the MD5 HMAC will be unavailable for use with SCTP. This option defaults to '`enabled`' on little-endian architectures, and '`disabled`' otherwise.

`--with sctp-adler32`
: Enable Adler32 checksum. This provides the ability to use the older RFC 2960 Adler32 checksum. If CONFIG_SCTP_CRC_32 below is not selected, the Adler32 checksum is always provided. This option defaults to '`disabled`'.

`--without sctp-crc32c`
: Disable CRC-32C checksum. This provides the ability to use the newer CRC-32c checksum as described in RFC 3309. When this is selected and CONFIG_SCTP_ADLER_32 is not selected above, then the only checksum that will be used is the CRC-32c checksum. This option defaults to '`enabled`'.

`--with sctp-throttle-passiveopens`
: Enable throttling of passive opens. Special feature of Linux SCTP not mentioned in RFC 2960. When secure algorithms are used for signing cookies, the implementation becomes vulnerable to INIT and `COOKIE-ECHO` flooding. If you enable this feature (`--with sctp-throttle-passiveopens`), SCTP will only allow one INIT and one `COOKIE-ECHO` to be processed in each interval corresponding to the sysctl sctp_throttle_itvl. Setting sctp_throttle_itvl to 0 defeats this function. If you disable this feature (`--without sctp-throttle-passiveopens`), each INIT and `COOKIE-ECHO` will be processed. This option defaults to '`disabled`'.

`--without sctp-ecn`
: Enable explicit congestion notification. This enables support for Explicit Congestion Notification (ECN) chunks in SCTP messages as defined in RFC 2960 and RFC 3168. It also adds syctl (/proc/net/ipv4/sctp_ecn) which allows ECN for SCTP to be disabled at runtime. This option defaults to '`enabled`'.

`--without sctp-lifetimes`
: Enable SCTP message lifetimes. This enables support for message lifetimes as described in RFC 2960. When enabled, message lifetimes can be set on messages. See sctp(7). This feature is always enabled when Partial Reliability Support is set. This option defaults to '`enabled`'.

`--without sctp-add-ip`
: Enable ADD-IP. This enables support for ADD-IP as described in draft-ietf-tsvwg-addip-sctp-07.txt. This allows the addition and removal of IP addresses from existing connections. This is experimental stuff. This option defaults to '`enabled`'.

`--without sctp-adaptation-layer-info`
: Enable ALI. This enables support for the Adaptation Layer Information parameter described in draft-ietf-tsvwg-addip-sctp-07.txt for communicating application layer information bits at initialization. This is experimental stuff. This option defaults to '`enabled`'.

`--without sctp-partial-reliability`
: Enable SCTP Partial Reliability (PR-SCTP). This enables support for PR-SCTP as described in draft-stewart-tsvwg-prsctp-03.txt. This allows for partial reliability of message delivery on a "timed reliability" basis. This is experimental stuff. This option defaults to '`enabled`'.

`--without sctp-error-generator`
: Disable the SCTP error generator. This provides an internal error generator that can be accessed with socket options for testing SCTP operation under packet loss. You will need this option to run some of the test programs distributed with the SCTP module. This option defaults to '`enabled`'.

`--without ip`
: Remove the second generation IP driver from the build. The default is to include the second generation IP driver in the build.

`--without udp`
: Remove the second generation UDP driver from the build. The default is to include the second generation UDP driver in the build.

`--without raw`
: Remove the second generation RAWIP driver from the build. The default is to include the second generation RAWIP driver in the build.

`--without tcp`
: Remove the second generation TCP driver from the build. The default is to include the second generation TCP driver in the build.

`--with sctp`
: Enable the version 1 driver in the build. This option defaults to '`disabled`'.

`--without sctp2`
: Enable the Release 2 driver in the build. This option defaults to '`enabled`'.

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 8.4.1 [Building from the Source RPM], page 169.

8.3.4 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 8.3.5 [Configuring the Tar Ball], page 143. For an example, see Section 8.4.2 [Building from the Debian DSC], page 170.

Build

To build from the Debian DSC, see Section 8.4.2 [Building from the Debian DSC], page 170.

8.3.5 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.

8.3.5.1 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)`.

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 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 useful 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 run-time 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 run-time and static compile components are desired, but not major documentation. This option does not override the setting of `--disable-devel`.

`--enable-distribute-docs`
: Enables the distribution of pre-built documentation. Distribution of pre-built documentation in the distribution tarball causes the tarball size to increase dramatically. To avoid this, the distribution of pre-built documentation was suppressed, by default, to reduce the size of the distribution tarball. Enabling this feature causes pre-built documentation to be included in the distribution tarball. The default is to disable the distribution of pre-built documentation.

`--disable-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(8)` 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.

`--disable-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(8)` or `dpkg(1)`. The '`rebuild`' `automake(1)` target uses this feature to rebuild for all available architectures and kernels.

`--disable-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.

`--disable-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.

`--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.

`--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`, `--enable-k-debug` and `--enable-k-none` below. This has the effect of invoking some more pedantic assertion macros in the code. The default is not to apply kernel safety.

`--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` and `--enable-k-none` below. This has the effect of removing `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.

`--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, and `--enable-k-none` below. 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.

`--enable-k-none`
: Enable no kernel module run-time checks. Specifies whether kernel run-time checks are to be performed. This option is mutually exclusive with `--enable-k-safe`, `--enable-k-test` and `--enable-k-debug`. This has the effect of removing all assertion macros from the code. The default is to not to remove all assertion macros.

`--disable-k-modversions`
: Disable module versions on `OpenSS7` symbols. Specifies whether kernel symbol versions are to be used on symbols exported from built `OpenSS7` modules. The default is to provide kernel symbol versions on all exported symbols.

`--enable-k-package`
: Enable generation of a kernel source package. Specifies whether the source package: installation of configured source in `/usr/src/OpenSS7`, will be generated. This also affects `rpm(8)` and `dpkg(1)` builds to include a source package. Generation of source pacakges is disabled by default.

`--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(8) 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(8) 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-patchlevel=PATCHLEVEL
: Specifies the patch level for the package. This is neither used for rpm(8) nor dpkg(1) packages, it applies to the tarball release as a whole. The default is the contents of the .pkgpatchlevel file in the release package source directory or, if that file does not exist, the null string. This is an additional suffix after the package release in the package version number. It is intended for bug fix releases or release candidates.

--with-pkg-distdir=DIR
: Specifies the distribution directory for the package. This is used by the maintainer for building distribution repositories. This is the directory into which binary packages are copied for distribution. The default is the top build directory.

--with-pkg-tardir=DIR
: Specifies the tarball 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.[26] 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.

[26] 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-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.

`--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-hdrdir=K-HEADER-DIR`
: Specify the '`K-INCLUDES-DIR`' include directory of the kernel for which the build is targeted. resides. The default is *`DESTDIR/K-BUILD-DIR`*`/include`. 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-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-asm=K-ASMDIR`
: Specifiy the '`K-ASMDIR`' kernel source architecture specific directory. The default is *`DESTDIR/K-HEADER-DIR`*`/asm`. 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 settings are *ricing* and the resulting kernel modules will only be about 5% faster.

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

`--disable-texinfo`
: Disable INFO-formatted texinfo documents. Normally `configure` can find the tools necessary to build INFO documents from texinfo and will provide missing substitutions instead. This option need only be provided when requested by `configure` script error messages. This option can otherwise be used to suppress the creation of texinfo INFO-formatted documents. The default is to enable INFO-formatted texinfo documents.

`--disable-texinfo-html`
: Disable HTML-formatted texinfo documents. Normally `configure` can find the tools necessary to build HTML documents from texinfo and will provide missing substitutions instead. This option need only be provided when requested by `configure` script error messages. This option can otherwise be used to suppress the creation of texinfo HTML-formatted documents. The default is to enable HTML-formatted texinfo documents.

`--disable-texinfo-print`
: Disable PS- and PDF-formatted texinfo documents. Normally `configure` can find the tools necessary to build PS and PDF documents from texinfo and will provide missing substitutions instead. This option need only be provided when requested by `configure` script error messages. This option can otherwise be used to suppress the creation of texinfo PS- and PDF-formatted documents. The default is to enable PS and PDF-formatted texinfo documents.

`--disable-papers`
: Disable build and install of latex papers. Normally `configure` can find the tools necessary to build TXT documents from latex and will provide missing substitutions instead. This option need only be provided when requested by `configure` script error messages. This option can otherwise be used to suppress the creation of latex TXT-formatted documents. The default is to automatically determine whether to build and install of latex papers.

`--disable-papers-html`
: Disable HTML-formatted latex papers. Normally `configure` can find the tools necessary to build HTML documents from latex and will provide missing substitutions instead. This option need only be provided when requested by `configure` script error messages. This option can otherwise be used to suppress the creation of latex HTML-formatted documents. The default is to automatically determine whether to enable HTML-formatted latex papers.

`--disable-papers-print`
: Disable PS- and PDF-formatted latex papers. Normally `configure` can find the tools necessary to build PS and PDF documents from latex and will provide missing substitutions instead. This option need only be provided when requested by `configure`

script error messages. This option can otherwise be used to suppress the creation of PS- and PDF-formatted documents. The default is to automatically determine whether to enable print-formatted latex papers.

`--disable-drafts`
: Disable build and install of TXT-formatted internet drafts. Normally `configure` can find the tools necessary to build TXT documents from groff ME and will provide missing substitutions instead. This option need only be provided when requested by `configure` script error messages. This option can otherwise be used to suppress the creation of TXT-formatted documents. The default is to automatically determine whether to enable build and install of TXT-formatted internet drafts.

`--disable-drafts-html`
: Disable HTML-formatted internet drafts. Normally `configure` can find the tools necessary to build HTML documents from groff ME and will provide missing substitutions instead. This option need only be provided when requested by `configure` script error messages. This option can otherwise be used to suppress the creation of HTML-formatted documents. The default is to automatically determine whether to enable HTML-formatted internet drafts.

`--disable-drafts-print`
: Disable PS- and PDF-formatted internet drafts. Normally `configure` can find the tools necessary to build PS and PDF documents from groff ME and will provide missing substitutions instead. This option need only be provided when requested by `configure` script error messages. This option can otherwise be used to suppress the creation of PS- and PDF-formatted documents. The default is to automatically determine whether to enable PS- and PDF-formatted internet drafts.

`--with-strconf-master=STRCONF_CONFIG`
: Specify the '`STRCONF_CONFIG`' file name to which the configuration master file is written. The default is `Config.master`.

`--with-base-major=STRCONF_MAJBASE`
: Start numbering for major devices at '`STRCONF_MAJBASE`'. The default is '`230`' on 16-bit device number systems, and '`2000`' on 32-bit device number systems.

`--with-base-modid=STRCONF_MODBASE`
: Start numbering for module identifiers at '`STRCONF_MODBASE`'. The default is '`5000`'.

`--with-strconf-pkgdir=PKG-DIRECTORY`
: Specifies the relative or absolute path to the binary package configuration directory in which to look for binary packages, '`PKG-DIRECTORY`'. The default is the `pkg` subdirectory in the build or source directory.

`--with-strconf-pkgrules=PKG-FILENAME`
: Specifies the relative or absolute file name to which package rules are written, '`PKG-FILENAME`'. The default is the `pkgrules` file in the build directory.

`--with-snmp-agent=SNMP-AGENT-HEADERS`
`--without-snmp-agent`
: Normally SNMP agents are included in the build. By specifying '`--without-snmp-agent`', the SNMP argents are not included in the build. This option is only for exceptional circumstances where SNMP support cannot be included in the run-time.

Otherwise, specifies the SNMP agent header file directory, '`SNMP-AGENT-HEADERS`'. The default is *`$INCLUDEDIR`* and the `configure` script will search for this directory. The 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-perl-headers HEADERS`
Normally `configure` can find the perl headers necessary for generating SNMP agents based on NET-SNMP. This option may be used to direct `configure` to the location of the perl headers. The default is '`$INCLUDEDIR/perl`'.

`--with-snmp=SNMP-HEADERS`
`--without-snmp`
Normally `configure` can find the NET-SNMP headers necessary for generating SNMP argents based on NET-SNMP. This option may be used to direct `configure` to the location of the NET-SNMP headers, '`SNMP-HEADERS`'. The default is '`$INCLUDEDIR/ucd-snmp`'. The 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.

`--disable-java`
Disable JAVA modules. The package contains JAVA modules that can only be built the the appropriate tool chain. The `configure` script can normally detect and recommend missing JAVA components that are needed to generate JAVA modules and documentation. This option should only be used on a system that does not support JAVA or when directed by a `configure` script error message. The default is to enable JAVA modules.

`--with-tcl=TCL-HEADERS`
`--without-tcl`
Specifies the TCL header file directory, '`TCL-HEADERS`'. The default is to search for the appropriate headers in a number of locations. The 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 contain TCL interfaces.

`--with-perl=PERL-HEADERS`
`--without-perl`
Specifies the PERL header file directory, '`PERL-HEADERS`'. The default is to search for the appropriate headers in a number of locations. The 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 contain PERL interfaces.

`--disable-bestzip`
`--enable-bestzip`
Disable best compression of archives. The `configure` script normally detects the available tools and determines what the best compression (highest compression ratio) is to compress `tar(1)` archives. It will typically use `.bz2` and `.xz` formats on current distributions but may fall back to `.gz` and `.bz2` compression on older distributions. The default is to enable the best compression of archives.

`--with-lzma=LZMA_CMD`
`--without-lzma`
: Specifies the creation of `lzma` archives. '`LZMA_CMD`' specifies the compression command to use (typically `lzma`). The typical use of is command is to specify `--without-lzma` to suppress the rather slow `lzma` compression for test compilations. The default is to create `lzma` archives when a suitable command can be found or was specified and no `xz` archive capability is provided.

  Unfortunately, `automake(1)` does not detect the presence of the necessary tools to build `lzma` archives, breaking distribution on systems that do not have `lzma` tools. This option tells `configure` to configure distribution to include `lzma` archives. As the generation of `lzma` takes a significant amount of time, generation of `lzma` archives is disabled by default.

`--with-xz=XZ_CMD`
`--without-xz`
: Specifies the creation of `xz` archives. '`XZ_CMD`' specifies the compression command to use (typically `xz`). The typical use of is command is to specify `--without-xz` to suppress the rather slow `lzma` compression for test compilations. The default is to create `xz` archives when a suitable command can be found or was specified.

  Unfortunately, `automake(1)` does not detect the presence of the necessary tools to build `xz` archives, breaking distribution on systems that do not have `xz` tools. This option tells `configure` to configure distribution to include `xz` archives. As the generation of `xz` takes a significant amount of time, generation of `xz` archives is disabled by default.

`--disable-repo-tar`
`--enable-repo-tar`
: Disable building of `tar(1)` repositories. Normally `configure` can determine whether there are sufficient tools to create `tar(1)` repositories. This option explicitly disables or enables generation of `tar(1)` repositories. Normally in maintainer mode, `tar(1)` repositories will be constructed. This option is normally used to disable the construction of `tar(1)` repositories.

`--disable-rpms`
`--enable-rpms`
: Disable building of RPMs. Normally `configure` can determine whether it is possible to build RPMs on the current system and automatically determines whether to build RPMs as part of the '`release`' makefile target. This option explicitly disables or enables building of RPMs. The default is to auotmatically determine whether RPMs should be built. This option is only enabled automatically when a suitable `rpm(8)` command can be located. A typical use of this option is to disable rpms with `--disable-rpms` on a *Debian* system that yet provides the `rpm(8)` commands.

`--disable-srpms`
`--enable-srpms`
: Disable building of SRPMs. Normally `configure` can determine whether it is possible to build SRPMs on the current system and automatically determines whether to build SRPMs as part of the '`release`' makefile target. This option explicitly disables or enables building of SRPMs. The default is to automatically determine whether SRPMs should be built. This option is only enabled automatically when a suitable `rpm(8)` command can be located. A typical use of this option is to disable rpms with `--disable-srpms` on a *Debian* system that yet provides the `rpm(8)` commands.

`--disable-repo-yum`
`--enable-repo-yum`
: Disable building of `yum(8)` repositories. Normally `configure` can determine whether there are sufficient tools to create `yum(8)` repositories. This option explicitly disables or enables generation of `yum(8)` repositories.

`--with-install-source-yum=REPODIR`
`--without-install-source-yum`
: Specifies the `yum(8)` repository directory, '`REPODIR`'. Creates and installs an installation source for the `yum(8)` package management tool. The default is to search for the appropriate directory.

`--disable-repo-yast`
`--enable-repo-yast`
: Disable building of `yast(8)` repositories. Normally `configure` can determine whether there are sufficient tools to create `yast(8)` repositories. This option explicitly disables or enables generation of `yast(8)` repositories.

`--with-install-source-zypp=REPODIR`
`--without-install-source-zypp`
: Specifies the `zypper(8)` repository directory, '`REPODIR`'. Creates and installs an installation source for the `zypper(8)` package management tool. The default is to search for the appropriate directory.

`--disable-debs`
`--enable-debs`
: Disable building of DEBs. Normally `configure` can determine whether it is possible to build DEBs on the current system and automatically determines whether to build DEBs as part of the '`release`' makefile target. This option explicitly disables or enable building of DEBs.

`--disable-dscs`
`--enable-dscs`
: Disable building of DSCs. Normally `configure` can determine whether it is possible to build DSCs on the current system and automatically determines whether to build DSCs as part of the '`release`' makefile target. This option explicitly disables or enable building of DSCs.

`--disable-repo-apt`
`--enable-repo-apt`
: Disable building of `apt-get(1)` repositories. Normally `configure` can determine whether there are sufficient tools to create `apt-get(1)` repositories. This option explicitly disables or enables generation of `apt-get(1)` repositories.

`--with-install-source-apt=SOURCESDIR`
`--without-install-source-apt`
: Specifies the `apt(1)` sources directory, '`SOURCESDIR`'. Creates and installs an installation source for the `aptitude(8)` package management tool. The default is to search for the appropriate directory.

The following `configure` options, specific to the OpenSS7, are available:

`--disable-streams-irq`
: Disables STREAMS irq suppression. Normally the STREAMS scheduler protects itself and its datastructures from races due to hard interrupts by suppressing interrupts

during critical sections. As not all drivers and modules contain hard interrupts, this option allows hard interrupts to be enabled while running critical sections. The purpose of this option was primarily for profiling. The default is to enable STREAMS irq suppression.

`--disable-streams-stats`
: Disable STREAMS statistics counting. Normally the STREAMS scheduler will automatically count the number of entries to open, close, put and service procedures for a queue pair whenever the `module_stats` structure is defined and attached to the `qinit` structure by the module or driver. This is not exact SVR4.2 MP behaviour (where it is the responsibility of the module or drive to perform these counts). This option disables the feature. The default is to enable STREAMS statistics counting.

`--disable-streams-syncqs`
: Disable STREAMS synchronization queues. Normally the STREAMS scheduler will permit modules and drivers that are written for syncrhonization (such as SVR4.2 MP synchronization, Solaris perimeters, etc.) and will perform synchronization protection for these modules. This option disables synchronization queues. When disabled, only fully multiprocessor safe drivers and modules (marked with the D_MP flag), will be loaded. The default is to enable STREAMS synchronization queues.

`--disable-streams-utils`
: Disable additional STREAMS utilities. Normally `strsetup` and `strload` utility configuration files are included in the build and installed. This option disables build and installation of the `strsetup` and `strload` configuration files. The default is to enable additional STREAMS utilities.

`--disable-big-compile`
: Disable compilation as one big computational unit. The default is to build as one big computational unit. Do not use this option.

`--enable-streams-fifos`
: Enable overriding of system FIFOs with STREAMS-based FIFOs. The default is to not override system FIFOs with STREAMS-based FIFOs.

`--with-streams-kthreads`
: Set STREAMS kernel thread operation to *nice*, *normal*, *rt* or *no*. When set to *nice*, the STREAMS scheduler will be based on kernel threads that run with a '`nice -19`' priority. When set to *normal*, the STREAMS scheduler sill be based on kernel threas thar run with '`nice -0`' priority. When set to *rt*, the STREAMS scheduler will be based on kernel threads that run with real-time priority '`nice -99`'. When set to *no*, soft-interrupts will be used for the STREAMS scheduler rather than kernel threads. This option was primarily for performance testing. Do no use this option. The default STREAMS kernel thread pirority is *nice*.

`--enable-module-sth`
`--disable-module-sth`
: Enables or disables the `sth` (Stream head) module. When enabled, the `sth` module will be compiled into the `streams` kernel module; when disabled, the `sth` module will not be included at all. Note that disabling the `sth` module will cause all Streams to fail. The default is for the `sth` module to be created as a separate module.

`--enable-module-srvmod`
`--disable-module-srvmod`
: Enables or disables the `srvmod` module. When enabled, the `srvmod` module will be compiled into the `streams` kernel module; when disabled, the `srvmod` module will not be included at all. Note that disabling the `srvmod` module will cause conformance suites to fail. The default is for the `srvmod` module to be created as a separate module.

`--enable-module-nullmod`
`--disable-module-nullmod`
: Enables or disables the `nullmod` module. When enabled, the `nullmod` module will be compiled into the `streams` kernel module; when disabled, the `nullmod` module will not be included at all. Note that disabling the `nullmod` module will cause conformance suites to fail. The default is for the `nuls` module to be created as a separate module.

`--enable-module-pipemod`
`--disable-module-pipemod`
: Enables or disables the `pipemod` module. When enabled, the `pipemod` module will be compiled into the `streams` kernel module; when disabled, the `pipemod` module will not be included at all. Note that disabling the `pipemod` module will cause conformance suites to fail. The default is for the `pipemod` module to be created as a separate module.

`--enable-module-connld`
`--disable-module-connld`
: Enables or disables the `connld` module. When enabled, the `connld` module will be compiled into the `streams` kernel module; when disabled, the `connld` module will not be included at all. Note that disabling the `connld` module will cause conformance suites to fail. The default is for the `connld` module to be created as a separate module.

`--enable-module-sc`
`--disable-module-sc`
: Enables or disables the `sc` module. When enabled, the `sc` module will be compiled into the `streams` kernel module; when disabled, the `sc` module will not be included at all. Note that disabling the `sc` module will cause conformance suites to fail. The default is for the `sc` module to be created as a separate module.

`--enable-module-testmod`
`--disable-module-testmod`
: Enables or disables the `testmod` module. When enabled, the `testmod` module will be compiled into the `streams` kernel module; when disabled, the `testmod` module will not be included at all. Note that disabling the `testmod` module will cause conformance suites to fail. The default is for the `testmod` module to be created as a separate module.

`--enable-module-timod`
`--disable-module-timod`
: Enables or disables the `timod` module. When enabled, the `timod` module will be compiled into the `streams` kernel module; when disabled, the `timod` module will not be included at all. Note that disabling the `timod` module will cause conformance suites to fail. The default is for the `timod` module to be created as a separate module.

`--enable-module-tirdwr`
`--disable-module-tirdwr`
: Enables or disables the `tirdwr` module. When enabled, the `tirdwr` module will be compiled into the `streams` kernel module; when disabled, the `tirdwr` module will not

be included at all. Note that disabling the `tirdwr` module will cause conformance suites to fail. The default is for the `tirdwr` module to be created as a separate module.

`--enable-module-bufmod`
`--disable-module-bufmod`
: Enables or disables the `bufmod` module. When enabled, the `bufmod` module will be compiled into the `streams` kernel module; when disabled, the `bufmod` module will not be included at all. Note that disabling the `bufmod` module will cause conformance suites to fail. The default is for the `bufmod` module to be created as a separate module.

`--enable-module-pfmod`
`--disable-module-pfmod`
: Enables or disables the `pfmod` module. When enabled, the `pfmod` module will be compiled into the `streams` kernel module; when disabled, the `pfmod` module will not be included at all. Note that disabling the `pfmod` module will cause conformance suites to fail. The default is for the `pfmod` module to be created as a separate module.

`--enable-module-nbuf`
`--disable-module-nbuf`
: Enables or disables the `nbuf` module. When enabled, the `nbuf` module will be compiled into the `streams` kernel module; when disabled, the `nbuf` module will not be included at all. Note that disabling the `nbuf` module will cause conformance suites to fail. The default is for the `nbuf` module to be created as a separate module.

`--enable-module-pf`
`--disable-module-pf`
: Enables or disables the `pf` module. When enabled, the `pf` module will be compiled into the `streams` kernel module; when disabled, the `pf` module will not be included at all. Note that disabling the `pf` module will cause conformance suites to fail. The default is for the `pf` module to be created as a separate module.

`--enable-driver-clone`
`--disable-driver-clone`
: Enables or disables the `clone` module. When enabled, the `clone` module will be compiled into the `streams` kernel module; when disabled, the `clone` module will not be included at all. Note that disabling the `clone` module will cause conformance suites to fail. The default is for the `clone` driver to be created as a separate module.

`--enable-driver-echo`
`--disable-driver-echo`
: Enables or disables the `echo` module. When enabled, the `echo` module will be compiled into the `streams` kernel module; when disabled, the `echo` module will not be included at all. Note that disabling the `echo` module will cause conformance suites to fail. The default is for the `echo` driver to be created as a separate module.

`--enable-driver-fifo`
`--disable-driver-fifo`
: Enables or disables the `fifo` module. When enabled, the `fifo` module will be compiled into the `streams` kernel module; when disabled, the `fifo` module will not be included at all. Note that disabling the `fifo` module will cause conformance suites to fail. The default is for the `fifo` driver to be created as a separate module.

`--enable-driver-log`
`--disable-driver-log`
: Enables or disables the `log` module. When enabled, the `log` module will be compiled into the `streams` kernel module; when disabled, the `log` module will not be included at all. Note that disabling the `log` module will cause conformance suites to fail. The default is for the `log` driver to be created as a separate module.

`--enable-driver-loop`
`--disable-driver-loop`
: Enables or disables the `loop` module. When enabled, the `loop` module will be compiled into the `streams` kernel module; when disabled, the `loop` module will not be included at all. Note that disabling the `loop` module will cause conformance suites to fail. The default is for the `loop` driver to be created as a separate module.

`--enable-driver-nsdev`
`--disable-driver-nsdev`
: Enables or disables the `nsdev` module. When enabled, the `nsdev` module will be compiled into the `streams` kernel module; when disabled, the `nsdev` module will not be included at all. Note that disabling the `nsdev` module will cause conformance suites to fail. The default is for the `nsdev` driver to be created as a separate module.

`--enable-driver-mux`
`--disable-driver-mux`
: Enables or disables the `mux` module. When enabled, the `mux` module will be compiled into the `streams` kernel module; when disabled, the `mux` module will not be included at all. Note that disabling the `mux` module will cause conformance suites to fail. The default is for the `mux` driver to be created as a separate module.

`--enable-driver-nuls`
`--disable-driver-nuls`
: Enables or disables the `nuls` module. When enabled, the `nuls` module will be compiled into the `streams` kernel module; when disabled, the `nuls` module will not be included at all. Note that disabling the `nuls` module will cause conformance suites to fail. The default is for the `nuls` driver to be created as a separate module.

`--enable-driver-pipe`
`--disable-driver-pipe`
: Enables or disables the `pipe` module. When enabled, the `pipe` module will be compiled into the `streams` kernel module; when disabled, the `pipe` module will not be included at all. Note that disabling the `pipe` module will cause conformance suites to fail. The default is for the `pipe` driver to be created as a separate module.

`--enable-driver-sad`
`--disable-driver-sad`
: Enables or disables the `sad` module. When enabled, the `sad` module will be compiled into the `streams` kernel module; when disabled, the `sad` module will not be included at all. Note that disabling the `sad` module will cause conformance suites to fail. The default is for the `sad` driver to be created as a separate module.

`--enable-driver-sfx`
`--disable-driver-sfx`
: Enables or disables the `sfx` module. When enabled, the `sfx` module will be compiled into the `streams` kernel module; when disabled, the `sfx` module will not be included

at all. Note that disabling the `sfx` module will cause conformance suites to fail. The default is for the `sfx` driver to be created as a separate module.

`--enable-driver-spx`
`--disable-driver-spx`
: Enables or disables the `spx` module. When enabled, the `spx` module will be compiled into the `streams` kernel module; when disabled, the `spx` module will not be included at all. Note that disabling the `spx` module will cause conformance suites to fail. The default is for the `spx` driver to be created as a separate module.

`--enable-compat-os7`
`--disable-compat-os7`
: Enables or disables the `os7` compatibility module. When enabled, the `os7` compatibility module will be compiled into the `stream` kernel module; when disabled, the `os7` compatibility module will not be included at all. The default is for the `os7` compatibility module to be created as a separate module.

`--enable-compat-svr3`
`--disable-compat-svr3`
: Enables or disables the `svr3` compatibility module. When enabled, the `svr3` compatibility module will be compiled into the `stream` kernel module; when disabled, the `svr3` compatibility module will not be included at all. The default is for the `svr3` compatibility module to be created as a separate module.

`--enable-compat-svr4`
`--disable-compat-svr4`
: Enables or disables the `svr4` compatibility module. When enabled, the `svr4` compatibility module will be compiled into the `stream` kernel module; when disabled, the `svr4` compatibility module will not be included at all. The default is for the `svr4` compatibility module to be created as a separate module.

`--enable-compat-mps`
`--disable-compat-mps`
: Enables or disables the `mps` compatibility module. When enabled, the `mps` compatibility module will be compiled into the `stream` kernel module; when disabled, the `mps` compatibility module will not be included at all. The default is for the `mps` compatibility module to be created as a separate module.

`--enable-compat-sol8`
`--disable-compat-sol8`
: Enables or disables the `sol8` compatibility module. When enabled, the `sol8` compatibility module will be compiled into the `stream` kernel module; when disabled, the `sol8` compatibility module will not be included at all. The default is for the `sol8` compatibility module to be created as a separate module.

`--enable-compat-uw7`
`--disable-compat-uw7`
: Enables or disables the `uw7` compatibility module. When enabled, the `uw7` compatibility module will be compiled into the `stream` kernel module; when disabled, the `uw7` compatibility module will not be included at all. The default is for the `uw7` compatibility module to be created as a separate module.

`--enable-compat-osf`
`--disable-compat-osf`
: Enables or disables the `osf` compatibility module. When enabled, the `osf` compatibility module will be compiled into the `stream` kernel module; when disabled, the `osf` compatibility module will not be included at all. The default is for the `osf` compatibility module to be created as a separate module.

`--enable-compat-aix`
`--disable-compat-aix`
: Enables or disables the `aix` compatibility module. When enabled, the `aix` compatibility module will be compiled into the `stream` kernel module; when disabled, the `aix` compatibility module will not be included at all. The default is for the `aix` compatibility module to be created as a separate module.

`--enable-compat-hpux`
`--disable-compat-hpux`
: Enables or disables the `hpux` compatibility module. When enabled, the `hpux` compatibility module will be compiled into the `stream` kernel module; when disabled, the `hpux` compatibility module will not be included at all. The default is for the `hpux` compatibility module to be created as a separate module.

`--enable-compat-irix`
`--disable-compat-irix`
: Enables or disables the `irix` compatibility module. When enabled, the `irix` compatibility module will be compiled into the `stream` kernel module; when disabled, the `irix` compatibility module will not be included at all. The default is for the `irix` compatibility module to be created as a separate module.

`--enable-compat-mac`
`--disable-compat-mac`
: Enables or disables the `mac` compatibility module. When enabled, the `mac` compatibility module will be compiled into the `stream` kernel module; when disabled, the `mac` compatibility module will not be included at all. The default is for the `mac` compatibility module to be created as a separate module.

`--disable-xti-servtype`
: Disables XTI service type checks in the XTI/TLI Library. Normally, the XTI/TLI Library will check for the service type of the endpoint and will reject commands that are not defined for the corresponding service type. When enabled, this option causes the XTI/TLI Library to simply issue the corresponding primitive to the underlying driver and to allow the driver to determine whether the primitive is supported. The default is for the XTI/TLI Library to check for XTI service type.

`--disable-xti-states`
: Disables XTI state checks in the XTI/TLI Library. Normally the XTI/TLI Library will check for the state of the endpoint and will reject commands that would place the interface out of state. When enabled, this option causes the XTI/TLI Library to simply issue the corresponding primitive to the underlying driver and to allow the driver to determine whether the interface is out of state. The default is for the XTI/TLI Library to check for XTI state.

`--enable-sctp-slow-verification`
: Enable slow verification of addresses and tags. When a message comes from an SCTP endpoint with the correct verification tag, it is not necessary to check whether it is from

a correct source address to identify the SCTP association to which it belongs. When you disable this feature (`--disable-sctp-slow-verification`), source addresses are not checked and it is up to firewall implementations to thwart attackers of the verification tag. When you enable this feature (`--enable-sctp-slow-verification`), you get RFC 2960 compliant operation, but at great cost to SCTP performance. This option defaults to '`disabled`'.

`--enable-sctp-throttle-heartbeats`
: Enable heartbeat throttling. Special feature of OpenSS7 that is not mentioned in RFC 2960. When you enable this feature (`--enable-sctp-throttle-heartbeats`), OpenSS7 will throttle the rate at which it responds to heartbeats to the system control *heartbeat_interval*. This makes SCTP more resilient to implementations which flood heartbeat messages. For RFC 2960 compliant operation, disable this feature (`--disable-sctp-throttle-heartbeats`). This option defaults to '`disabled`'.

`--enable-sctp-discard-ootb`
: Enable discard of out-of-the-blue packets. RFC 2960 requires the implementation to send `ABORT` to some OOTB packets (packets for which no SCTP association exists). Sending `ABORT` chunks to unverified source addresses with the T bit set opens SCTP to blind masquerade attacks. Not sending them may lead to delays at the peer endpoint aborting associations where our `ABORT` has been lost and the socket is already closed or if we have restarted and the peer still has open associations to us. If you enable this feature (`--enable-sctp-discard-ootb`), SCTP will discard all OOTB packets. This is necessary if another SCTP stack is being run on the same machine. Therefore, if the OpenSS7 package is included on an OpenSS7 SCTP kernel, this feature is automatically enabled. For RFC 2960 compliant operation, disable this feature (`--disable-sctp-discard-ootb`). This option defaults to '`disabled`' for non-OpenSS7 SCTP kernels, and '`enabled`' for OpenSS7 SCTP kernels.

`--enable-sctp-extended-ip-support`
: Enable extended IP support for SCTP. This provides extended IP support for SCTP for things like IP Transparent Proxy and IP Masquerading. This is experimental stuff. If in doubt, disable this feature (`--disable-sctp-extended-ip-support`). This option defaults to '`disabled`'.

`--disable-sctp-hmac-sha1`
: Disable SHA-1 HMAC. This provides the ability to use the FIPS 180-1 (SHA-1) message authentication code in SCTP cookies. If you enable this feature (`--enable-sctp-hmac-sha1`), when the appropriate sysctl is set, SCTP will use the SHA-1 HMAC when signing cookies in the `INIT-ACK` chunk. If disable this feature (`--disable-sctp-hmac-sha1`), the SHA-1 HMAC will be unavailable for use with SCTP. This option defaults to '`enabled`' on big-endian architectures, and '`disabled`' otherwise.

`--disable-sctp-hmac-md5`
: Disable MD5 HMAC. This provides the ability to use the MD5 (RFC 1321) message authentication code in SCTP cookies. If you enable this feature (`--enable-sctp-hmac-md5`), when the appropriate sysctl is set, SCTP will use the MD5 HMAC when signing cookies in the INIT ACK chunk. If you disable this feature (`--disable-sctp-hmac-md5`), the MD5 HMAC will be unavailable for use with SCTP. This option defaults to '`enabled`' on little-endian architectures, and '`disabled`' otherwise.

`--enable-sctp-adler32`
: Enable Adler32 checksum. This provides the ability to use the older RFC 2960 Adler32 checksum. If CONFIG_SCTP_CRC_32 below is not selected, the Adler32 checksum is always provided. This option defaults to '`disabled`'.

`--disable-sctp-crc32c`
: Disable CRC-32C checksum. This provides the ability to use the newer CRC-32c checksum as described in RFC 3309. When this is selected and CONFIG_SCTP_ADLER_32 is not selected above, then the only checksum that will be used is the CRC-32c checksum. This option defaults to '`enabled`'.

`--enable-sctp-throttle-passiveopens`
: Enable throttling of passive opens. Special feature of Linux SCTP not mentioned in RFC 2960. When secure algorithms are used for signing cookies, the implementation becomes vulnerable to INIT and `COOKIE-ECHO` flooding. If you enable this feature (`--enable-sctp-throttle-passiveopens`), SCTP will only allow one INIT and one `COOKIE-ECHO` to be processed in each interval corresponding to the sysctl sctp_throttle_itvl. Setting sctp_throttle_itvl to 0 defeats this function. If you disable this feature (`--disable-sctp-throttle-passiveopens`), each INIT and `COOKIE-ECHO` will be processed. This option defaults to '`disabled`'.

`--disable-sctp-ecn`
: Enable explicit congestion notification. This enables support for Explicit Congestion Notification (ECN) chunks in SCTP messages as defined in RFC 2960 and RFC 3168. It also adds syctl (/proc/net/ipv4/sctp_ecn) which allows ECN for SCTP to be disabled at runtime. This option defaults to '`enabled`'.

`--disable-sctp-lifetimes`
: Enable SCTP message lifetimes. This enables support for message lifetimes as described in RFC 2960. When enabled, message lifetimes can be set on messages. See sctp(7). This feature is always enabled when Partial Reliability Support is set. This option defaults to '`enabled`'.

`--disable-sctp-add-ip`
: Enable ADD-IP. This enables support for ADD-IP as described in draft-ietf-tsvwg-addip-sctp-07.txt. This allows the addition and removal of IP addresses from existing connections. This is experimental stuff. This option defaults to '`enabled`'.

`--disable-sctp-adaptation-layer-info`
: Enable ALI. This enables support for the Adaptation Layer Information parameter described in draft-ietf-tsvwg-addip-sctp-07.txt for communicating application layer information bits at initialization. This is experimental stuff. This option defaults to '`enabled`'.

`--disable-sctp-partial-reliability`
: Enable SCTP Partial Reliability (PR-SCTP). This enables support for PR-SCTP as described in draft-stewart-tsvwg-prsctp-03.txt. This allows for partial reliability of message delivery on a "timed reliability" basis. This is experimental stuff. This option defaults to '`enabled`'.

`--disable-sctp-error-generator`
: Disable the SCTP error generator. This provides an internal error generator that can be accessed with socket options for testing SCTP operation under packet loss. You will

need this option to run some of the test programs distributed with the SCTP module. This option defaults to '`enabled`'.

`--without-ip`
Remove the second generation IP driver from the build. The default is to include the second generation IP driver in the build.

`--without-udp`
Remove the second generation UDP driver from the build. The default is to include the second generation UDP driver in the build.

`--without-raw`
Remove the second generation RAWIP driver from the build. The default is to include the second generation RAWIP driver in the build.

`--without-tcp`
Remove the second generation TCP driver from the build. The default is to include the second generation TCP driver in the build.

`--with-sctp`
Enable the version 1 driver in the build. This option defaults to '`disabled`'.

`--without-sctp2`
Enable the Release 2 driver in the build. This option defaults to '`enabled`'.

8.3.5.2 Environment Variables

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

CC C compiler command. Defaults to `gcc`.

CFLAGS C compiler flags. Defaults to an automatically determined set of flags.

LDFLAGS Linker flags. Defaults to an automatically determined set of flags.

CPPFLAGS
C/C++/Objective C preprocessor flags. Defaults to an automatically determined set of flags.

CPP C preprocessor. Defaults to `gcc -E`.

CCAS Assembler compiler command (defaults to '`CC`').

CCASFLAGS
Assembler compiler flags (defaults to CFLAGS). Defaults to an automatically determined set of flags.

CXX C++ compiler command. Defaults to `g++`.

CXXFLAGS
C++ compiler flags. Defaults to an automatically determined set of flags.

CXXCPP C++ preprocessor. Defaults to `g++ -E`.

LD Linker loader command. Defaults to `gcc`.

YACC The 'Yet Another C Compler' implementation to use. Defaults to the first program found out of: 'bison -y', 'byacc', 'yacc'.

YFLAGS The list of arguments that will be passed by default to '`$YACC`'. This script will default '`YFLAGS`' to the empty string to avoid a default value of `-d` given by some make applications.

SWIG Swig command. Defaults to `swig`.

GCJ Java compiler command. Defaults to `gcj`.

GCJFLAGS
Java complier flags. Defaults to an automatically determined set of flags.

GCJDBTOOL
GCJ database tool. Defaults to `gcj-dbtool`.

GCJH Java CNI header command. Defaults to `gcjh`.

GCJHFLAGS
Java CNI header command flags. Defaults to an automatically determined set of flags.

JAVAH Java JNI header command. Defaults to `gcjh`.

JAVAHFLAGS
Java JNI header command flags. Defaults to an automatically determined set of flags.

JAVAC Java class compiler. Defaults to `gcj`.

JAVACFLAGS
Java class compiler flags. Defaults to an automatically determined set of flags.

CLASSPATH
Java CLASSPATH variable. Defaults to *auto*.

JAVADOC Java documentation doclet. Defaults to `gjdoc`.

JAVADOCFLAGS
Java documentation flags. Defaults to an automatically determined set of flags.

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`. The default is '`--best`'.

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.

LZMA Default compression options given to `LZMA_CMD` is '`--best`'.

LZMA_CMD
Lzma compression command. Defaults to `lzma`.

XZ Default compression options given to `XZ_CMD` is '`-f9v`'.

XZ_CMD Xz compression command. Defaults to `xz`.

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

JAR Java archive command. Defaults to this first found of `fastjar`, or `jar`.

ZIP Zip archive command. Defaults to `zip`.

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.

INSSERV Insert service command. Defautls to `insserv`.

RPM Rpm command, `rpm(8)`. 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(8)` will be used instead of `rpmbuild(1)` only if `rpmbuild(1)` cannot be found.

CREATEREPO
Create `repo-md` repository command. This command is used when building `yum(8)` repositories. By default, `configure` will search for this tool.

MODIFYREPO
: Modify `repo-md` respository command. This command is used when building `yum(8)` repositories. By default, `configure` will search for this tool.

CREATE_PACKGE_DESCR
: Create YaST package descriptions command. This command is used when building `yast(8)` repositories. By default, `configure` will search for this tool.

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.

APT_FTPARCHIVE
: Apt-ftparchive command, `apt-ftparchive(1)`. This command is used to building Debian apt repositories. By default, `configure` will search for this tool.

DPKG_SCANSOURCES
: Dpkg-scansources command, `dpkg-scansources(1)`. This command is used to create the sources file when building Debian apt repositories. By default, `configure` will search for this tool.

DPKG_SCANPAKCAGES
: Dpkg-scanpackages command, `dpkg-scanpackages(1)`. This command is used to create the packages file when building Debian apt repositories. By default, `configure` will search for this tool.

DPKG_DEB
: Dpkg-deb command, `dpkg-deb(1)`. This command is used when building Debian apt repositories. By default, `configure` will search for this tool.

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 and 3.x 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.

DOXYGEN
Doxygen command, `doxygen(1)`. This command is used when building Doxygen documentation. By default, `configure` will search for this tool.

TEX Tex command for PS. Defaults to `tex`, `etex`. By default, `configure` will search for this tool.

PDFTEX Tex command for PDF. Defaults to `pdftex`, `pdfetex`. By default, `configure` will search for this tool.

BIBTEX BibTeX command. Defaults to `bibtex`. By default, `configure` will search for this tool.

LATEX Latex command. Defaults to `latex`. By default, `configure` will search for this tool.

PSLATEX PS Latex command. Defaults to `pslatex`. By default, `configure` will search for this tool.

PDFLATEX
PDF Latex command. Defaults to `pdflatex`. By default, `configure` will search for this tool.

LATEX2HTML
: LaTeX to HTML command. Defaults to `latex2html`. By default, `configure` will search for this tool.

DVI2PS
: DVI to PS conversion command. Defaults to `dvips`. By default, `configure` will search for this tool.

DVIPDF
: DVI to PDF conversion command. Defatuls to `dvipdf`. By default, `configure` will search for this tool.

PS2PDF
: PS to PDF conversion command. Defaults to `ps2pdf`. By default, `configure` will search for this tool.

GNUPLOT
: GNU plot command. Defaults to `gnuplot`. By default, `configure` will search for this tool.

GROFF
: Roff formatting command. Default `groff`. By default, `configure` will search for this tool.

FIG2DEV
: Fig to graphics format command. Defaults to `fig2dev`. By default, `configure` will search for this tool.

CONVERT
: Graphics format conversion command. Defaults to `convert`. By default, `configure` will search for this tool.

PS2EPSI
: PS to EPSI conversion command. Defaults to `ps2epsi`. By default, `configure` will search for this tool.

EPSTOPDF
: EPS to PDF conversion command. Defaults to `epstopdf`. By default, `configure` will search for this tool.

MD5SUM
: MD5 sum command, `md5sum(1)`. This command is used to checksum tarballs when creating FTP archive respositories. By default, `configure` will search for this tool. It normally defaults to `md5sum`.

SHA1SUM
: SHA1 sum command, `sha1sum(1)`. This command is used to checksum tarballs when creating FTP archive respositories. By default, `configure` will search for this tool. It normally defaults to `sha1sum`.

SHA256SUM
: SHA256 sum command, `sha256sum(1)`. This command is used to checksum tarballs when creating FTP archive respositories. By default, `configure` will search for this tool. It normally defaults to `sha256sum`.

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.

8.3.5.3 Build

To build from the tar ball, see Section 8.4.3 [Building from the Tar Ball], page 170.

8.4 Building

The *OpenSS7* can be built from source RPM, Debian DSC, or tarball. Any of these approaches requires access to source code. Note, however, that you must be entitled to receive source code to be able to build using any of these techniques. *OpenSS7* does not normally permit access to source code beginning with the 1.1.1 release. Only sponsors of the *OpenSS7 Project* and some clients of *OpenSS7 Corporation* are permitted access to source.

8.4.1 Building from the Source RPM

If you have downloaded the necessary source RPM (see Section 8.2.7 [Downloading the Source RPM], page 127), 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 8.5.5 [Installing the Binary RPM], page 175).
The source RPM is rebuilt to binary RPMs as follows:

```
% wget https://www.openss7.org/repo/rpms/SRPMS/openss7-1.1.7.20141001-1.src.rpm
% rpmbuild --rebuild -vv openss7-1.1.7.20141001-1.src.rpm
```

The rebuild process can also recognize a number of options that can be used to tweak the resulting binaries, see Section 8.3.3 [Configuring the Source RPM], page 132. These options are provided on the `rpm(8)` command line. For example:

```
% rpmbuild --rebuild -vv --target athlon-redhat-linux \
  --define "_kversion 3.0.99-1-unx" \
  -- openss7-1.1.7.20141001-1.src.rpm
```

will rebuild binary RPM for the '`3.0.99-1-unx`' kernel for the '`athlon`' architecture.[27]

[27] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example.

Installation

To install the resulting binary RPM, see Section 8.5.5 [Installing the Binary RPM], page 175.

8.4.2 Building from the Debian DSC

If you have downloaded the necessary Debian DSC (see Section 8.2.8 [Downloading the Debian DSC], page 127), 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 8.5.6 [Installing the Debian DEB], page 175).

The Debian DSC is rebuilt to binary DEBs as follows:

```
% wget http://www.openss7.org/debian/openss7_1.1.7.20141001-0.dsc
% wget http://www.openss7.org/debian/openss7_1.1.7.20141001-0.tar.gz
% dpkg-buildpackage -v openss7_1.1.7.20141001-0.dsc
```

The rebuild process can also recognize a number of options that can be used to tweak the resulting binaries, see Section 8.3.4 [Configuring the Debian DSC], page 143. These options are provided in the environment variable *BUILD_DPKGOPTIONS* and have the same form as the options to `configure`, see Section 8.3.5 [Configuring the Tar Ball], page 143. For example:

```
% BUILD_DEBOPTIONS='
        --with-k-release=3.0.99-1-unx
        --host=athlon-debian-linux-gnu'
  dpkg-buildpackage -v \
  openss7_1.1.7.20141001-0.dsc
```

will rebuild binary DEB for the '`3.0.99-1-unx`' kernel for the '`athlon`' architecture.[28]

Installation

To install the resulting binary DEB, see Section 8.5.6 [Installing the Debian DEB], page 175.

8.4.3 Building from the Tar Ball

If you have downloaded the tar ball (see Section 8.2.9 [Downloading the Tar Ball], page 127), then the following instructions will rebuild the package on your system. (Note that the build process does not required `root` privilege.)

8.4.3.1 Native Build

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

```
% wget https://www.openss7.org/repo/tarballs/openss7-1.1.7.20141001.tar.xz
% tar -xJvf openss7-1.1.7.20141001.tar.xz
% pushd openss7-1.1.7.20141001
% ./configure
% make
% popd
```

[28] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example.

8.4.3.2 Cross-Build

Following is an example for a cross-build. The kernel release version must always be specified for a cross-build.[29] 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 https://www.openss7.org/repo/tarballs/openss7-1.1.7.20141001.tar.xz
% tar -xJvf openss7-1.1.7.20141001.tar.xz
% pushd openss7-1.1.7.20141001
% ./configure DESTDIR="/some/other/root" \
--with-k-release=2.4.18 --host sparc-linux
% make
% popd
```

8.5 Installing

8.5.1 Installing with YUM

If you have set up the necessary repository definitions (see Section 8.1.2.1 [Setting up YUM], page 112), then the following instructions will install the RPMs on your system.[30] The repository includes groups (patterns) and virtual packages that ease the installation and removal of kernel modules, libraries and utilities.

To install the *OpenSS7* run-time components for a single-kernel distribution installation, use (one of):

```
% sudo yum install openss7
% sudo yum install @openss7
% sudo yum groupinstall openss7
```

To include the *OpenSS7 JAIN* Java components, use (one of):

```
% sudo yum install openss7-java
% sudo yum install @openss7-java
% sudo yum groupinstall openss7-java
```

To add an additional *OpenSS7* kernel run-time component to the installation, use (one of):

```
% sudo yum install openss7-$(uname -r)
% sudo yum install openss7-3.0.99-1-unx
```

Here, '`$(uname -r)`' is simply a way of specifying the running kernel. The value '`3.0.99-1-unx`' is just an example. Use the specific kernel version that you want to add to the installation.

[29] 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.

[30] For additional information on `yum(8)`, see `yum(8)`.

To install the *OpenSS7* development components for a single-kernel distribution, use (one of):

```
% sudo yum install openss7-devel
% sudo yum install @openss7-devel
% sudo yum groupinstall openss7-devel
```

To add an additional *OpenSS7* kernel development component to the installation, use (one of):

```
% sudo yum install openss7-devel-$(uname -r)
% sudo yum install openss7-devel-3.0.99-1-unx
```

Here, '`$(uname -r)`' is simply a way of specifying the running kernel. The value '`3.0.99-1-unx`' is just an example. Use the specific kernel version that you want to add to the installation.

To add *OpenSS7* development documentation, use (one of):

```
% sudo yum install openss7-doc openss7-javadoc
% sudo yum install @openss7-doc openss7-javadoc
% sudo yum groupinstall openss7-doc
```

8.5.2 Installing with ZYPPER

If you have set up the necessary repository definitions (see Section 8.1.2.4 [Setting up ZYPPER], page 113), then the following instructions will install the RPMs on your system.[31] The repository includes groups (patterns) and virtual packages that ease the installation and removal of kernel modules, libraries and utilities.

To install the *OpenSS7* run-time components for a single-kernel distribution installation, use (one of):

```
% sudo zypper install openss7
% sudo zypper install -t pattern openss7
```

To include the *OpenSS7 JAIN* Java components, use (one of):

```
% sudo zypper install openss7-java
% sudo zypper install -t pattern openss7-java
```

To add an additional *OpenSS7* kernel run-time component to the installation, use (one of):

```
% sudo zypper install openss7-$(uname -r)
% sudo zypper install openss7-3.0.99-1-unx
```

Here, '`$(uname -r)`' is simply a way of specifying the running kernel. The value '`3.0.99-1-unx`' is just an example. Use the specific kernel version that you want to add to the installation.

To install the *OpenSS7* development components for a single-kernel distribution, use (one of):

[31] For additional information on `zypper(8)`, see `zypper(8)`.

```
% sudo zypper install openss7-devel
% sudo zypper install -t pattern openss7-devel
```

To add an additional *OpenSS7* kernel development component to the installation, use (one of):

```
% sudo zypper install openss7-devel-$(uname -r)
% sudo zypper install openss7-devel-3.0.99-1-unx
```

Here, '`$(uname -r)`' is simply a way of specifying the running kernel. The value '`3.0.99-1-unx`' is just an example. Use the specific kernel version that you want to add to the installation.

To add *OpenSS7* development documentation, use (one of):

```
% sudo zypper install openss7-doc openss7-javadoc
% sudo zypper install -t pattern openss7-doc
```

8.5.3 Installing with URPMI

If you have set up the necessary repository definitions (see Section 8.1.4.1 [Setting up URPMI], page 116), then the following instructions will install the RPMs on your system.[32] The repository includes groups (patterns) and virtual packages that ease the installation and removal of kernel modules, libraries and utilities.

To install the *OpenSS7* run-time components for a single-kernel distribution installation, use (one of):

```
% sudo urpmi openss7
```

To include the *OpenSS7 JAIN* Java components, use (one of):

```
% sudo urpmi openss7-java
```

To add an additional *OpenSS7* kernel run-time component to the installation, use (one of):

```
% sudo urpmi openss7-$(uname -r)
% sudo urpmi openss7-3.0.99-1-unx
```

Here, '`$(uname -r)`' is simply a way of specifying the running kernel. The value '`3.0.99-1-unx`' is just an example. Use the specific kernel version that you want to add to the installation.

To install the *OpenSS7* development components for a single-kernel distribution, use (one of):

```
% sudo urpmi openss7-devel
```

To add an additional *OpenSS7* kernel development component to the installation, use (one of):

```
% sudo urpmi openss7-devel-$(uname -r)
```

[32] For additional information on `urpmi(8)`, see `urpmi(8)`.

Here, '`$(uname -r)`' is simply a way of specifying the running kernel. The value '`3.0.99-1-unx`' is just an example. Use the specific kernel version that you want to add to the installation.

To add *OpenSS7* development documentation, use (one of):

```
% sudo urpmi openss7-doc openss7-javadoc
```

8.5.4 Installing with APT

If you have set up the necessary repository definitions (see Section 8.1.5.1 [Setting up APT], page 117), then the following instructions will install the DEBs on your system.[33] The repository includes groups (patterns) and virtual packages that ease the installation and removal of kernel modules, libraries and utilities.

To install the *OpenSS7* run-time components for a single-kernel distribution installation, use (one of):[34]

```
% sudo apt-get install openss7
% sudo aptitude install openss7
```

To include the *OpenSS7 JAIN* Java components, use (one of):

```
% sudo apt-get install openss7-java
% sudo aptitude install openss7-java
```

To add an additional *OpenSS7* kernel run-time component to the installation, use (one of):

```
% sudo apt-get install openss7-$(uname -r)
% sudo apt-get install openss7-3.0.99-1-unx
% sudo aptitude install openss7-$(uname -r)
% sudo aptitude install openss7-3.0.99-1-unx
```

Here, '`$(uname -r)`' is simply a way of specifying the running kernel. The value '`3.0.99-1-unx`' is just an example. Use the specific kernel version that you want to add to the installation.

To install the *OpenSS7* development components for a single-kernel distribution, use (one of):

```
% sudo apt-get install openss7-devel
% sudo aptitude install openss7-devel
```

To add an additional *OpenSS7* kernel development component to the installation, use (one of):

```
% sudo apt-get install openss7-devel-$(uname -r)
% sudo aptitude install openss7-devel-$(uname -r)
```

Here, '`$(uname -r)`' is simply a way of specifying the running kernel. The value '`3.0.99-1-unx`' is just an example. Use the specific kernel version that you want to add to the installation.

[33] For additional information on `apt(8)`, see `apt(8)`.

[34] Note that `aptitude(8)` is not always available on systems supporting APT-RPM.

To add *OpenSS7* development documentation, use (one of):

```
% sudo apt-get install openss7-doc openss7-javadoc
% sudo aptitude install openss7-doc openss7-javadoc
```

8.5.5 Installing the Binary RPM

If you have downloaded the necessary binary RPMs (see Section 8.2.5 [Downloading the Binary RPM], page 121), or have rebuilt binary RPMs using the source RPM (see Section 8.4.1 [Building from the Source RPM], page 169), then the following instructions will install the RPMs on your system. For additional information on `rpm(8)`, see `rpm(8)`.

```
% pushd RPMS/i686
% rpm -ihv openss7-*-1.1.7.20141001-1.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(8)`, see `rpm(8)`. For example, the following will relocate the documentation and info directories:

```
% pushd RPMS/i686
% rpm -ihv \
       --relocate '/usr/share/doc=/usr/local/share/doc' \
       --relocate '/usr/share/info=/usr/local/share/info' \
       -- openss7-doc-1.1.7.20141001-1.i686.rpm
```

The previous example will install the `openss7-doc` package by will relocate the documentation an info directory contents to the `/usr/local` version.

8.5.6 Installing the Debian DEB

If you have downloaded the necessary Debian DEBs (see Section 8.2.6 [Downloading the Debian DEB], page 125), or have rebuild binary DEBs using the Debian DSC (see Section 8.4.2 [Building from the Debian DSC], page 170), then the following instructions will install the DEBs on your system. For additional information see `dpkg(8)`.

```
% pushd debian
% dpkg -iv openss7-*_1.1.7.20141001-0_*.deb
```

You must have the correct `.deb` files downloaded or build for this to be successful.[35]

8.5.7 Installing the Tar Ball

After the build process (see Section 8.4.3 [Building from the Tar Ball], page 170), installation only requires execution of one of two `automake(1)` targets:

[35] Depending on the release, and if you have used option '`--enable-maintainer-mode`' to `configure` and have generated DEBs with '`make release`', you should be able to issue '`make install-debs`' from the build directory.

'`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.

8.6 Removing

8.6.1 Removing with YUM

OpenSS7 repositories support `yum(8)` in `repo-md` XML format. The repository includes virtual packages that ease the removal of kernel modules, libraries and utilities.

To remove all *OpenSS7* components, use (one of):

```
$> sudo yum remove openss7
$> sudo yum remove @openss7
$> sudo yum groupremove openss7
```

To remove just the *OpenSS7 JAIN* Java component, use (one of):

```
$> sudo yum remove openss7-java
$> sudo yum remove @openss7-java
$> sudo yum groupremove openss7-java
```

To remove the *OpenSS7* run-time components for a specific kernel, use (one of):

```
$> sudo yum remove openss7-$(uname -r)
$> sudo yum remove openss7-3.0.99-1-unx
```

Where, '`$(uname -r)`' will remove for the running kernel. '`3.0.99-1-unx`' is just an example. Note also that *OpenSS7* does not install a kernel package for each and every kernel version, nor is it normally necessary. When the kernel run-time being removed is the last kernel run-time, the remainder of the *OpenSS7* components will also be removed.

To remove all *OpenSS7* development components, use (one of):

```
$> sudo yum remove openss7-develop
$> sudo yum remove @openss7-develop
$> sudo yum groupremove openss7-develop
```

To remove the *OpenSS7* development components for a specific kernel, use (one of):

```
$> sudo yum remove openss7-develop-$(uname -r)
$> sudo yum remove openss7-develop-3.0.99-1-unx
```

Where, '`$(uname -r)`' will remove for the running kernel. '`3.0.99-1-unx`' is just an example. When the kernel development component being removed is the last kernel development component, the remainder of the *OpenSS7* development components will also be removed.

To remove just documentation components, use (one of):

```
$> sudo yum remove openss7-doc openss7-javadoc
$> sudo yum remove @openss7-doc
$> sudo yum groupremove openss7-doc
```

8.6.2 Removing with ZYPPER

OpenSS7 repositories support `zypper(8)` in SUSE `repo-md` XML format. The repository includes virtual packages that ease the removal of kernel modules, libraries and utilities.

To remove all *OpenSS7* components, use (one of):

```
$> sudo zypper remove openss7
```

To remove just the *OpenSS7 JAIN* Java component, use (one of):

```
$> sudo zypper remove openss7-java
```

To remove the *OpenSS7* run-time components for a specific kernel, use (one of):

```
$> sudo zypper remove openss7-$(uname -r)
$> sudo zypper remove openss7-3.0.99-1-unx
```

Where, '`$(uname -r)`' will remove for the running kernel. '`3.0.99-1-unx`' is just an example. Note also that *OpenSS7* does not install a kernel package for each and every kernel version, nor is it normally necessary. When the kernel run-time being removed is the last kernel run-time, the remainder of the *OpenSS7* components will also be removed.

To remove all *OpenSS7* development components, use (one of):

```
$> sudo zypper remove openss7-develop
```

To remove the *OpenSS7* development components for a specific kernel, use (one of):

```
$> sudo zypper remove openss7-develop-$(uname -r)
$> sudo zypper remove openss7-develop-3.0.99-1-unx
```

Where, '`$(uname -r)`' will remove for the running kernel. '`3.0.99-1-unx`' is just an example. When the kernel development component being removed is the last kernel development component, the remainder of the *OpenSS7* development components will also be removed.

To remove just documentation components, use (one of):

```
$> sudo zypper remove openss7-doc openss7-javadoc
```

8.6.3 Removing with URPMI

OpenSS7 repositories support `urpmi(8)` in `hdlist` format. The repository includes virutal packages that ease the removal of kernel modules, libraries and utilities.
To remove all *OpenSS7* components, use (one of):

```
$> sudo urpme openss7
```

To remove just the *OpenSS7 JAIN* Java component, use (one of):

```
$> sudo urpme openss7-java
```

To remove the *OpenSS7* run-time components for a specific kernel, use (one of):

```
$> sudo urpme openss7-$(uname -r)
$> sudo urpme openss7-3.0.99-1-unx
```

Where, '`$(uname -r)`' will remove for the running kernel. '`3.0.99-1-unx`' is just an example. Note also that *OpenSS7* does not install a kernel package for each and every kernel version, nor is it normally necessary. When the kernel run-time being removed is the last kernel run-time, the remainder of the *OpenSS7* components will also be removed.
To remove all *OpenSS7* development components, use (one of):

```
$> sudo urpme openss7-develop
```

To remove the *OpenSS7* development components for a specific kernel, use (one of):

```
$> sudo urpme openss7-develop-$(uname -r)
$> sudo urpme openss7-develop-3.0.99-1-unx
```

Where, '`$(uname -r)`' will remove for the running kernel. '`3.0.99-1-unx`' is just an example. When the kernel development component being removed is the last kernel development component, the remainder of the *OpenSS7* development components will also be removed.
To remove just documentation components, use (one of):

```
$> sudo urpme openss7-doc openss7-javadoc
```

Of course, you are welcome to use a GUI package manager, such as `rpmdrake(8)`.

8.6.4 Removing with APT

8.6.5 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 '*openss7*') 2>&1 | tee erase.log
```

This should uninstall all installed OpenSS7 RPMs from your system.[36] Note that this command may also be used for any system based on rpm(8) regardless of whether the RPMs where installed using rpm(8), yum(8) or zypper(8).
For more information see rpm(8).

8.6.6 Removing the Debian DEB

To remove an 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 '^openss7') 2>&1 | tee erase.log
```

This should uninstall all installed OpenSS7 DEBs from your system.[37] Note that this command may also be used for any system based on dpkg(1) regardless of whether the DEBs were installed using dpkg(1) or apt(8).
For more information see dpkg(1).

8.6.7 Removing the Source RPM

To remove all the installed binary RPM build from the source RPM, see Section 8.6.5 [Removing the Binary RPM], page 178. Then simply remove the binary RPM package files and source RPM file. A command such as:

```
% find / -name 'openss7-*.rpm' -type f -print0 | xargs --null rm -f
```

This should remove all OpenSS7 RPMs from your system.[38]

8.6.8 Removing the Debian DSC

To remove all the installed binary DEB build from the Debian DSC, see Section 8.6.6 [Removing the Debian DEB], page 179. Then simply remove the binary DEB package files and Debian DSC file. A command such as:

```
% find / \( -name 'openss7-*.deb' \
         -o -name 'openss7-*.dsc' \
         -o -name 'openss7-*.tar.* \
         \) -type f -print0 | xargs --null rm -f
```

This should remove all OpenSS7 DEBs, DSCs and TARs from your system.[39]

8.6.9 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:

[36] Depending on the release, and if you have used option '`--enable-maintainer-mode`' to `configure` and have generated RPMs with '`make release`', you should be able to issue '`make uninstall-rpms`' from the build directory to remove all installed binary rpms belonging to the build release.

[37] Depending on the release, and if you have used option '`--enable-maintainer-mode`' to `configure` and have generated DEBs with '`make release`', you should be able to issue '`make uninstall-debs`' from the build directory to remove all installed Debian debs belonging to the build release.

[38] Depending on the release, you might be able to issue '`make remove-rpms`' from the build directory.

[39] Depending on the release, you might be able to issue '`make remove-debs`' from the build directory.

```
% cd /usr/src/openss7
% make uninstall
% cd ..
% rm -fr openss7-1.1.7.20141001
% rm -f openss7-1.1.7.20141001.tar.bz2
% rm -f openss7-1.1.7.20141001.tar.xz
```

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 8.5.7 [Installing the Tar Ball], page 175) except the final installation and then perform the steps above.

8.7 Loading

8.7.1 Normal Module Loading

When OpenSS7 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/3.0.99-1-unx/openss7`. The '`make install`' process should have copied the kernel module files `streams-*.o` to the directory `/lib/modules/3.0.99-1-unx/openss7`. This means that to load any of these modules, you can simply execute, for example, '`modprobe stream-`*`somedriver`*'.[40]

8.7.1.1 Linux Fast-STREAMS Module Loading

The `openss7` demand load system supports both the old `kerneld` and the new `kmod` mechanisms for demand loading kernel modules.

The convention for `openss7` kernel loadable object files is:

- Their name start with "streams-".
- They are placed in `/lib/modules/3.0.99-1-unx/streams/`, where '`3.0.99-1-unx`' is an example kernel version.

If your kernel has been built using the `kerneld` daemon, then `OpenSS7` 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.openss7
if -f /lib/modules/$(uname -r)/modules.openss7
include /lib/modules/$(uname -r)/modules.openss7
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.openss7` 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.

[40] Note that the '`_kversion`' of '`3.0.99-1-unx`' is only an example.

Linux Fast-STREAMS has a wider range of kernel module loading mechanisms than previously provided. For mechanisms used for kernel module loading under *Linux Fast-STREAMS*, See *Linux Fast-STREAMS Reference Manual*.

8.8 Maintenance

8.8.1 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:

8.8.1.1 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 9.1.1 [Pre-installation Checks], page 191.

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(8)` 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/openss7`[41] 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.[42]

  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.[43]

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

[41] `/usr/libexec/openss7` 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).

[42] 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.

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

8.8.1.2 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 `openss7-1.1.7.20141001.tar.bz2` and `openss7-1.1.7.20141001.tar.xz`.

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.

8.8.1.3 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.[44] It also means that you probably need a full blown Linux system to rebuild the package. For more information, see Section 8.2.10 [Downloading from CVS], page 128.

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.

8.8.1.4 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.

8.8.1.5 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. 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(8)` 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.

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

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 *GNUPG-PASS* 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.

8.8.1.6 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.[45] Following are the logging targets:

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`'.

[45] 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)`.

'`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`.

8.8.1.7 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).

8.8.1.8 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:

`openss7-1.1.7.20141001.tar.bz2` and `openss7-1.1.7.20141001.tar.xz`

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:

`openss7-1.1.7.20141001.tar.bz2.asc` and `openss7-1.1.7.20141001.tar.xz.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.

8.8.1.9 RPM Build Targets

On `rpm(8)` systems, or systems sporting rpm packaging tools, the following targets are used to generate `rpm(8)` 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: `openss7-1.1.7.20141001-1.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:

`openss7-*-1.1.7.20141001-1.*.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.

8.8.1.10 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: `openss7_1.1.7.20141001-0.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: `openss7-*_1.1.7.20141001-0_*.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.

8.8.1.11 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)` documentation 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`.

9 Troubleshooting

9.1 Test Suites

9.1.1 Pre-installation Checks

Most *OpenSS7* packages, including the *OpenSS7* 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 8.4.1 [Building from the Source RPM], page 169; and Section 8.4.2 [Building from the Debian DSC], page 170), 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 8.4.3 [Building from the Tar Ball], page 170; and Section 8.5.7 [Installing the Tar Ball], page 175), 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 9.1.

```
% wget http://www.openss7.org/openss7-1.1.7.20141001.tar.bz2
% tar -xjvf openss7-1.1.7.20141001.tar.bz2
% pushd openss7-1.1.7.20141001
% ./configure
% make
% make check  # <------- invoke pre-installation checks
% popd
```

Example 9.1: *Invoking Pre-Installation Checks*

Pre-installation checks fall into two categories: *System Checks* and *Maintenance Checks*.

9.1.1.1 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 9.2 will perform *System* checks.

```
% wget http://www.openss7.org/openss7-1.1.7.20141001.tar.bz2
% tar -xjvf openss7-1.1.7.20141001.tar.bz2
% pushd openss7-1.1.7.20141001
% ./configure
% make
% make check  # <------ invokes System pre-installation checks
% popd
```

Example 9.2: *Invoking System Checks*

9.1.1.2 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 9.3 will perform *Maintenance* checks.

```
% wget http://www.openss7.org/openss7-1.1.7.20141001.tar.bz2
% tar -xjvf openss7-1.1.7.20141001.tar.bz2
% pushd openss7-1.1.7.20141001
% ./configure --enable-maintainer-mode
% make
% make check  # <------ invokes Maintenance pre-installation checks
% popd
```

Example 9.3: *Invoking Maintenance Checks*

9.1.1.3 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`'.

9.1.2 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.

9.1.2.1 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 9.4.

[1] This particular check has caught some name space pollution that has occurred in the 2.6.11 kernel.

```
% wget http://www.openss7.org/openss7-1.1.7.20141001.tar.bz2
% tar -xjvf openss7-1.1.7.20141001.tar.bz2
% pushd openss7-1.1.7.20141001
% ./configure
% make
% make check  # <------ invokes System pre-installation checks
% make install
% sudo make installcheck # <------- invokes post-installation tests
% popd
```

Example 9.4: *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 9.2 [Problem Reports], page 195, below.

Typical steps for invoking an installed `testsuite` standalone are shown in Example 9.5.

```
% [sudo] /usr/libexec/openss7/testsuite
```

Example 9.5: *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 9.2.4 [Stand Alone Problem Reports], page 198.

9.2 Problem Reports

9.2.1 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* 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)`).

9.2.2 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* 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 8.8.1.7 [Problem Report Targets], page 186. An example of how simple it is to generate a problem report is illustrated in Example 9.6.

[2] ‘`send-pr`’ launches the user’s *EDITOR* to edit the problem report before submitting it.

```
% 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 9.6: *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.

9.2.3 Automatic Problem Reports

The *OpenSS7* 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 8.8.1.6 [Logging Targets], page 185) is invoked, if the primary target fails, the `send-pr` shell script is invoked to automatically generate a problem report file suitable for the corresponding target (as described above under see Section 9.2.1 [Problem Report Guidelines], page 195). An example is shown in Example 9.7.

```
% 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 9.7: *Problem Report from Failed Logging Target*

9.2.4 Stand Alone Problem Reports

The *OpenSS7* package installs the `send-pr` script and its configuration file `send-pr.config` in `${libexecdir}/openss7` along with the validation `testsuite`, see See Section 9.1 [Test Suites], page 191. 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 9.8.

```
% [sudo] /usr/libexec/openss7/testsuite
% # test cases failed...
% /usr/libexec/openss7/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/openss7/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/openss7/send-pr --file problem.pr
```

Example 9.8: *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`.

9.3 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* package.

Unverified behaviour may contain unknown bugs.

Please remember that there is **NO WARRANTY**.

See also Section 7.5 [Bugs], page 80, or file `BUGS` in the release directory.

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/.

GNU Free Documentation License

GNU FREE DOCUMENTATION LICENSE
Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful 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.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

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 (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

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 fewer than five), unless they release you from this requirement.

   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", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

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. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

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 to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

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, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

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, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

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, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
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 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Indices

Index of Concepts

L

M

O

P

Q

R

S

T

U

W

Y

Z

Index of Data Types

Index of Functions and Macros

Index of Variables and Constants

M

N

O

P

R

S

T

X

Y

Z

Index of Files and Programs

P

S

T

Index of Configuration Options

O

P

R

S

T

U

X

Index of Makefile Targets

Index of Manual Pages Referenced

E

F

G

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z