

Linux Fast-STREAMS

Installation and Reference Manual

Version 0.7a Edition 4
Updated 2006-01-04
Package streams-0.7a.4

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

Copyright © 2001-2005 OpenSS7 Corporation <<http://www.openss7.com/>>
Copyright © 1997-2000 Brian F. G. Bidulock <bidulock@openss7.org>
All Rights Reserved.

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

This is texinfo edition 4 of the Linux Fast-STREAMS documentation, and is consistent with streams 0.7a. This manual was developed under the **OpenSS7 Project** and was funded in part by **OpenSS7 Corporation**.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the same conditions as for modified versions.

Short Contents

Acknowledgements	1
1 Introduction	3
2 Objective	5
3 Reference	7
4 Conformance	9
5 Releases	13
6 Installation	17
7 Troubleshooting	51
A Copying	57
B Documentation License	63
Indices	71

Table of Contents

Acknowledgements	1
Sponsors	1
Contributors	1
1 Introduction	3
1.1 Notice	3
1.2 Overview	3
1.3 Organization of this Guide	3
1.4 Conventions and Definitions	3
2 Objective	5
3 Reference	7
3.1 Files	7
3.2 Drivers	7
3.3 Modules	7
3.4 Libraries	8
3.5 Utilities	8
3.6 Development	8
4 Conformance	9
4.1 STREAMS Compatibility	9
4.2 Porting	10
5 Releases	13
5.1 Prerequisites	13
5.2 Compatibility	13
5.2.1 GNU/Linux Distributions	13
5.2.2 Kernel	14
5.2.3 Linux STREAMS	14
5.3 Release Notes	14
5.3.1 Release streams-0.7a.4	14
5.3.2 Release streams-0.7a.3	15
5.3.3 Release streams-0.7a-3	15
5.3.4 Release streams-0.7a-2	15
5.3.5 Release streams-0.7a-1	15
5.4 Bugs	16
5.5 Schedule	16
5.6 History	16

6	Installation	17
6.1	Downloading	17
6.1.1	Downloading the Binary RPM	17
6.1.2	Downloading the Debian DEB	19
6.1.3	Downloading the Source RPM	21
6.1.4	Downloading the Debian DSC	21
6.1.5	Downloading the Tar Ball	21
6.1.6	Downloading from CVS	22
6.2	Configuration	23
6.2.1	Configuring the Binary RPM	24
6.2.2	Configuring the Debian DEB	25
6.2.3	Configuring the Source RPM	25
6.2.4	Configuring the Debian DSC	31
6.2.5	Configuring the Tar Ball	31
6.2.5.1	Configure Options	31
6.2.5.2	Environment Variables	38
6.2.5.3	Build	41
6.3	Building	41
6.3.1	Building from the Source RPM	41
6.3.2	Building from the Debian DSC	42
6.3.3	Building from the Tar Ball	43
6.3.3.1	Native Build	43
6.3.3.2	Cross-Build	43
6.3.3.3	NexusWare Build	43
6.4	Installing	46
6.4.1	Installing the Binary RPM	46
6.4.2	Installing the Debian DEB	47
6.4.3	Installing the Tar Ball	47
6.5	Removing	47
6.5.1	Removing the Binary RPM	47
6.5.2	Removing the Debian DEB	47
6.5.3	Removing the Source RPM	48
6.5.4	Removing the Debian DSC	48
6.5.5	Removing the Tar Ball	48
6.6	Loading	49
6.6.1	Normal Module Loading	49
6.6.1.1	Linux STREAMS Module Loading	49
6.6.1.2	Linux Fast-STREAMS Module Loading	49
6.6.2	NexusWare Module Loading	50

7	Troubleshooting	51
7.1	Test Suites.....	51
7.1.1	Pre-installation Checks.....	51
7.1.1.1	Pre-Installation System Checks.....	51
7.1.1.2	Pre-Installation Maintenance Checks.....	52
7.1.1.3	Specific Pre-Installation Checks	52
7.1.2	Post-installation Checks.....	54
7.1.2.1	Running Test Suites.....	55
7.2	Problem Reports.....	55
7.3	Known Bugs.....	56
Appendix A	Copying.....	57
A.1	GNU General Public License	57
A.1.1	Preamble	57
A.1.2	Terms and Conditions for Copying, Distribution and Modification.....	58
A.1.3	How to Apply These Terms to Your New Programs	62
Appendix B	Documentation License.....	63
B.1	GNU Free Documentation License.....	63
B.1.1	Preamble	63
B.1.2	Terms and Conditions for Copying, Distribution and Modification.....	63
B.1.3	How to use this License for your documents	69
Indices	71	
	Index of Concepts.....	72
	Index of Data Types	74
	Index of Functions and Macros.....	75
	Index of Variables and Constants.....	76
	Index of Files and Programs	77
	Index of Configuration Options.....	78

Acknowledgements

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

Sponsors

Funding for completion of the Linux Fast-STREAMS package was provided in part by:

- OpenSS7 Corporation

Additional funding for **The OpenSS7 Project** was provided by:

- OpenSS7 Corporation
- Lockheed Martin
- Performance Technologies
- Motorola
- HOB International
- Comverse
- Sonus Networks
- France Telecom
- SS8 Networks
- Nortel Networks
- Verisign

Contributors

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

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

1 Introduction

This manual documents the design, implementation, installation, operation and future development schedule of the Linux Fast-STREAMS package.

1.1 Notice

This package is released and distributed under the *GNU General Public License* (see [Section A.1 \[GNU General Public License\]](#), page 57). Please note, however, that there are different licensing terms for the manual pages and some of the documentation (derived from X/Open publications and other sources). Consult the permission notices contained in the documentation for more information.

This guide, is released under the *GNU Free Documentation License* (see [Appendix B \[Documentation License\]](#), page 63) with all sections invariant.

1.2 Overview

This manual documents the design, implementation, installation, operation and future development of the Linux Fast-STREAMS package.

1.3 Organization of this Guide

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

Chapter 1 [Introduction] , page 3.	This introduction
Chapter 2 [Objective] , page 5.	Objective of the package
Chapter 3 [Reference] , page 7.	Contents of the package
Chapter 4 [Conformance] , page 9.	Conformance of the package
Chapter 5 [Releases] , page 13.	Releases of the package
Chapter 6 [Installation] , page 17.	Installation of the package
Chapter 7 [Troubleshooting] , page 51.	Troubleshooting of the package

1.4 Conventions and Definitions

This manual uses *texinfo* typographic conventions.

2 Objective

Linux Fast-STREAMS is intended as a high-performance, production replacement for *Linux STREAMS (LiS)*. *Linux Fast-STREAMS* has the following features in contrast to *Linux STREAMS*:

Linux Fast-STREAMS

LiS

Provided and optimized for Linux.

Maintains portability across Linux, User Space, QNX, DOS and other ports.

Promotes mainstream kernel adoption.

Places portability concerns before mainstream kernel adoption.

Lindented and follows normal kernel formatting and coding practices.

Non-lindented, does not follow normal kernel formatting and coding practices.

Provides compatibility modes for AIX, HPUNIX, OSF, Solaris, UnixWare, SVR 4.2 and LiS. Supports all major SVR 4.2 variants.

Provides some compatibility interfaces but also many Linux-derived interfaces specific to LiS.

Licensed under GPL with commercial licensing available.

Dubiously licensed under LGPL (yet parts are GPL).

Supports full SVR 4.2 ES/MP synchronization models.

Supports only single, Linux derived synchronization models.

Runs at SoftIRQ. By running at SoftIRQ, Linux Fast-STREAMS ensures that its place within the scheduling network-based events is appropriate for the Linux kernel.

Separate kernel threads.

Provides common SVR 4.2 system tunable parameters and system controls as well as /proc filesystem for support and debugging.

Provides only log-based debugging.

Provides support for SVR 4.2 STREAMS and other utilities.

Provides limited set of LiS only Linux-derived utilities.

Provides full named-streams device and device filesystem support.

Provides only character-based node support.

3 Reference

3.1 Files

```
'specfs.o'  
'streams.o'  
'streams-aixcompat.o'  
'streams-hpuxcompat.o'  
'streams-liscompat.o'  
'streams-osfcompat.o'  
'streams-suncompat.o'  
'streams-svr4compat.o'  
'streams-uw7compat.o'
```

3.2 Drivers

The configuration of *STREAMS* drivers and modules is performed when compiling the **Linux Fast-STREAMS** subsystem. The *STREAMS* subsystem, core drivers and modules are part of every **Linux Fast-STREAMS** system.

The following sections list the core drivers and modules, *STREAMS* kernel tunable parameters, and *STREAMS* configuration information.

```
'streams-clone.o ("clone")'  
    Clone device driver.  
  
'streams-echo.o ("echo")'  
    Echo (loopback) device driver.  
  
'streams-fifo.o ("fifo")'  
    FIFO (Named Pipe) driver.  
  
'streams-log.o ("strlog")'  
    STREAMS log driver.  
  
'streams-nsdev.o ("nsdev")'  
    Named STREAMS device driver.  
  
'streams-nuls.o ("nuls")'  
    Null stream driver.  
  
'streams-pipe.o ("pipe")'  
    Pipe driver.  
  
'streams-sad.o ("sad")'  
    STREAMS Administrative Driver.
```

3.3 Modules

The configuration of *STREAMS* drivers and modules is performed when compiling the **Linux Fast-STREAMS** subsystem. The *STREAMS* subsystem, core drivers and modules are part of every **Linux Fast-STREAMS** system.

The following sections list the core drivers and modules, *STREAMS* kernel tunable parameters, and *STREAMS* configuration information.

`'streams-connld.o ("connld")'`

Connld module.

`'streams-pipemod.o ("pipemod")'`

Pipe module.

`'streams-sc.o ("sc")'`

STREAMS configuration module.

`'streams-sth.o ("sth")'`

Stream Head module.

Additional modules are provided by add-on packages.

3.4 Libraries

3.5 Utilities

3.6 Development

For development using the *streams* package, See [section “About This Manual” in *STREAMS Programmer’s Guide*](#).

4 Conformance

4.1 STREAMS Compatibility

Linux Fast-STREAMS provides some degree of compatibility with other *STREAMS* implementation as follows:

— *SVR 4.2 ES/MP*

Linux Fast-STREAMS provides some degree of operational compatibility with *SVR 4.2 ES/MP* to ease portability and common comprehension, see [section “SVR 4.2 Compatibility” in *STREAMS Programmer’s Guide*](#).

— *AIX 5L Version 5.1*

Linux Fast-STREAMS provides some degree of operational compatibility with *AIX 5L Version 5.1* to ease portability and common comprehension, see [section “AIX Compatibility” in *STREAMS Programmer’s Guide*](#).

— *HP-UX 11.0i v2*

Linux Fast-STREAMS provides some degree of operational compatibility with *HP-UX 11.0i v2* to ease portability and common comprehension, see [section “HP-UX Compatibility” in *STREAMS Programmer’s Guide*](#).

— *OSF/1 1.2/Digital UNIX/True 64*

Linux Fast-STREAMS provides some degree of operational compatibility with *OSF/1 1.2/Digital UNIX* to ease portability and common comprehension, see [section “OSF/1 Compatibility” in *STREAMS Programmer’s Guide*](#).

— *UnixWare 7.1.3 (OpenUnix 8)*

Linux Fast-STREAMS provides some degree of operational compatibility with *UnixWare 7.1.3 (OpenUnix 8)* to ease portability and common comprehension, see [section “UnixWare Compatibility” in *STREAMS Programmer’s Guide*](#).

— *Solaris 9/SunOS 5.9*

Linux Fast-STREAMS provides some degree of operational compatibility with *Solaris 9/SunOS 5.9* to ease portability and common comprehension, see [section “Solaris Compatibility” in *STREAMS Programmer’s Guide*](#).

— *SUPER-UX*

Linux Fast-STREAMS provides some degree of operational compatibility with *SUPER-UX* to ease portability and common comprehension, see [section “SUX Compatibility” in *STREAMS Programmer’s Guide*](#).

— *UXP/V*

Linux Fast-STREAMS provides some degree of operational compatibility with *UXP/V* to ease portability and common comprehension, see [section “UXP Compatibility” in *STREAMS Programmer’s Guide*](#).

— *LiS-2.16.18*

Linux Fast-STREAMS provides some degree of operational compatibility with *LiS 2.16* to ease portability and common comprehension, see [section “LiS Compatibility” in *STREAMS Programmer’s Guide*](#).

For additional details, see [section “About This Manual” in *STREAMS Programmer’s Guide*](#).

4.2 Porting

— SVR 4.2 ES/MP

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *SVR 4.2 ES/MP* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from SVR 4.2 MP” in *Linux Fast-STREAMS Porting Guide*.

— AIX 5L Version 5.1

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *AIX 5L Version 5.1* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from AIX 5L Version 5.1” in *Linux Fast-STREAMS Porting Guide*.

— HP-UX 11.0i v2

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *HP-UX 11.0i v2* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from HP-UX 11.0i v2” in *Linux Fast-STREAMS Porting Guide*.

— OSF/1 1.2/Digital UNIX/True 64

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *OSF/1 1.2/Digital UNIX/True 64* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from OSF/1 1.2/Digital UNIX” in *Linux Fast-STREAMS Porting Guide*.

— UnixWare 7.1.3 (OpenUnix 8)

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *UnixWare 7.1.3 (OpenUnix 8)* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from UnixWare 7.1.3 (OpenUnix 8)” in *Linux Fast-STREAMS Porting Guide*.

— Solaris 9/SunOS 5.9

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *Solaris 9/SunOS 5.9* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from Solaris 9/SunOS 5.9” in *Linux Fast-STREAMS Porting Guide*.

— SUPER-UX

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *SUPER-UX* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from SUPER-UX” in *Linux Fast-STREAMS Porting Guide*.

— *UXP/V*

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *UXP/V* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from *UXP/V*” in *Linux Fast-STREAMS Porting Guide*.

— *LiS-2.16.18*

Linux Fast-STREAMS provides compatibility functions for source level compatibility with *LiS-2.16.18* and to ease porting of modules and drivers to *Linux Fast-STREAMS*. Portability considerations are maintained in a separate manual: see section “Porting from Linux STREAMS (LiS) 2.16.18” in *Linux Fast-STREAMS Porting Guide*.

For additional details, see section “About This Manual” in *Linux Fast-STREAMS Porting Guide*.

5 Releases

This is the OpenSS7 Release of the Linux Fast-STREAMS core, tools, drivers and modules that implement the *Linux Fast-STREAMS* SVR 4.2 MP STREAMS utility for Linux. This package is intended as a replacement package for *Linux STREAMS (LiS)*.

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

5.1 Prerequisites

Prerequisites for the Linux Fast-STREAMS package are as follows:

- A fairly LSB compliant GNU/Linux distribution.¹
- Linux 2.4 or 2.6 kernel (2.4.10 - 2.4.27) or (2.6.3 - 2.6.10)
- glibc2 or better.
- GNU info (for info files).
- GNU groff (for man pages).²

5.2 Compatibility

This section discusses compatibility with major prerequisites.

5.2.1 GNU/Linux Distributions

Linux Fast-STREAMS is compatible with the following *Linux* distributions:

- RedHat Linux 7.2 (RH7)
- RedHat Linux 7.3 (RH7)
- Performance Technologies *NexusWare24*
- RedHat Linux 8.0 (RH8)
- RedHat Linux 9 (RH9)
- SuSE 8.0 Professional
- Fedora Core 1 (FC1)
- Debian 3.0r2 Woody
- Mandrakelinux 9.2 (MDK92)
- RedHat Enterprise Linux 3.0 (EL3)
- WhiteBox Enterprise Linux 3.0 (WBEL3)
- CentOS Enterprise Linux 3.4 (centos34)
- Fedora Core 2 (FC2)
- SuSE 9.1 Personal
- Mandrakelinux 10.0 (MDK100)

¹ See [Section 5.2.1 \[GNU/Linux Distributions\]](#), page 13, 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.

- SuSE 9.2 Professional (SuSE9.2)
- Mandrakelinux 10.1 (MDK101)
- Fedora Core 3 (FC3)
- RedHat Enterprise Linux 4 (EL4)
- CentOS Enterprise Linux 4.0 (centos4)
- WhiteBox Enterprise Linux 4 (WBEL4)
- Fedora Core 4 (FC4)
- Lineox 4.026 (LEL4)
- Mandriva Linux LE2005 (MDK102)
- Performance Technologies NexusWare 8.0
- Debian 3.1r0a Sarge (untested)
- OpenSuSE 10.0 (untested)

When installing from the tarball (see [Section 6.4.3 \[Installing the Tar Ball\]](#), page 47), this distribution is probably compatible with a much broader array of distributions than those listed above. These are the distributions against which the current maintainer creates and tests builds.

5.2.2 Kernel

The *Linux Fast-STREAMS* package compiles as a *Linux* kernel module. It is not necessary to patch the *Linux* kernel to build or use the package.³ Nor do you have to recompile your kernel to build or use the package. OpenSS7 packages use `autoconf` scripts to adapt the package source to your existing kernel. The package builds and runs nicely against production kernels from the distributions listed above. Rather than relying on kernel versions, the `autoconf` scripts interrogate the kernel for specific features and variants to better adapt to distribution production kernels that have had patches applied over the official [kernel.org](#) sources.

The *Linux Fast-STREAMS* package is compatible with 2.4 kernel series after 2.4.10. It has been tested up to and including 2.6.11.

5.2.3 Linux STREAMS

Linux Fast-STREAMS provides a suitable replacement for the (now deprecated) *Linux STREAMS (LiS) 2.18.0* package formerly maintained by Dave Goethe of [GCOM](#).

5.3 Release Notes

The sections that follow provide information on OpenSS7 releases of the Linux Fast-STREAMS package.

5.3.1 Release streams-0.7a.4

This is primarily a bug fixes release and corrections resulting from testing.

³ 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.

5.3.2 Release streams-0.7a.3

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 re-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 an internal beta test release and was not released publicly.

5.3.3 Release streams-0.7a-3

Updates to common build process. Documentation updates.

This was an internal alpha test release and was not released publicly.

5.3.4 Release streams-0.7a-2

Removed all XTI/TLI and Linux networking code, headers and documentation from streams distribution and set epoch at 0. Linux networking code has been migrated to the *strxnet*, *strinet* and *strsctp* packages. The purpose for doing this was to allow the Linux networking to build against *Linux Fast-STREAMS* as well as *Linux STREAMS* and is a preparation for phasing out LiS and phasing in LfS.

This was an internal alpha test release and was not released publicly.

5.3.5 Release streams-0.7a-1

This is the initial release of the Linux Fast-STREAMS package for Linux. This is intended as a high-performance, production replacement for *Linux STREAMS (LiS)*. Linux Fast-STREAMS has the following features:

- optimized for Linux kernels.
- prepared for mainstream Linux kernel adoption.
- indented and follows kernel coding practices.
- compatibility modes for AIX, HP-UX, OSF, Solaris, UnixWare, SVR 4.2 and LiS.
- supports all major SVR4.2 variants.
- licensed under GPL with commercial licensing available.
- supports full SVR 4.2 MP synchronization models.
- runs at SoftIRQ.
- provides common SVR 4.2 system tunable parameters and sysctls.
- provides `/proc` filesystem access for debugging and performance tuning.
- provides a full set of common STREAMS modules and drivers.
- provides full name-streams device and shadow special filesystem support.

This was an internal alpha test release and was not released publicly.

5.4 Bugs

Linux Fast-STREAMS has many known bugs. These are alpha releases. Use at your own risk. Remember that there is **NO WARRANTY**.⁴

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

5.5 Schedule

5.6 History

⁴ See section **NO WARRANTY** under [Section A.1 \[GNU General Public License\]](#), page 57.

6 Installation

6.1 Downloading

The Linux Fast-STREAMS 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 **streams** from the links in the sections that follow.

By far the easiest form for installing and using **streams-0.7a.4** is to download and install binary RPM. If a binary RPM is not available for your distribution, but your distribution supports RPM, the next best method for installing and using **streams-0.7a.4** is to download and rebuild the source RPM. If your architecture does not support RPM at all, or you have special needs (such as cross-compiling for embedded targets), the final resort method is to download, configure, build and install from the source tarball.

6.1.1 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.

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 '**streams-source-0.7a.4-1.7.x.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 see [Section 6.3.1 \[Building from the Source RPM\]](#), page 41).

Architecture Independent

streams-dev-0.7a.4-1.7.x.noarch.rpm

The '**streams-dev**' package contains the device definitions necessary to run applications programs developed for Linux Fast-STREAMS.¹

streams-doc-0.7a.4-1.7.x.noarch.rpm

The '**streams-doc**' package contains this manual in plaintext, postscript, PDF and HTML forms, along with the meta-information from the '**streams**' package. It also contains all of the manual pages necessary for developing Linux Fast-STREAMS applications and Linux Fast-STREAMS STREAMS modules or drivers.

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

streams-init-0.7a.4-1.7.x.noarch.rpm

The ‘**streams-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.

streams-source-0.7a.4-1.7.x.noarch.rpm

The ‘**streams-source**’ package contains the source code necessary for building the Linux Fast-STREAMS release. It includes the **autoconf** configuration utilities necessary to create and distribute tarballs, rpms and deb/dscs.

Architecture Dependent

streams-devel-0.7a.4-1.7.x.i686.rpm

The ‘**streams-devel**’ package contains library archives for static compilation, header files to develop Linux Fast-STREAMS modules and drivers. This also includes the header files and static libraries required to compile Linux Fast-STREAMS applications programs.

streams-lib-0.7a.4-1.7.x.i686.rpm

The ‘**streams-lib**’ package contains the run-time shared libraries necessary to run application programs and utilities developed for the ‘**streams**’ package.

streams-util-0.7a.4-1.7.x.i686.rpm

The ‘**streams-util**’ package provides administrative and configuration test utilities and commands associated with the Linux Fast-STREAMS package.

Kernel-Dependent RPM

Kernel-Dependent RPM are dependent on specific Linux Kernel Binary RPM releases. Packages are provided for popular released *RedHat* kernels. Packages dependent upon *Red-Hat* or other kernel RPM will have the ‘**_kversion**’ kernel package version in the package name.

One of the following Kernel-Dependent packages is required for your architecture and kernel version. If your architecture or kernel version is not on the list, you can build binary RPM from the source RPM (see see [Section 6.3.1 \[Building from the Source RPM\]](#), page 41).²

streams-core-2.4.20-28.7bigmem-0.7a.4-1.7.x.i686.rpm

The ‘**streams-core**’ package contains the loadable kernel modules that depend only on the kernel. This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version ‘**2.4.20-28.7bigmem**’.³

² 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.

³ Note that the ‘**_kversion**’ of ‘**2.4.20-28.7bigmem**’ is only an example. Note also that only release packages that contain kernel modules will contain a ‘**core**’ subpackage.

streams-info-2.4.20-28.7bigmem-0.7a.4-1.7.x.i686.rpm

The ‘**streams-info**’ package⁴ contains the module symbol version information for the ‘**core**’ subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loading the actual kernel modules (from the ‘**core**’ subpackage above). This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version ‘2.4.20-28.7bigmem’.⁵

Configuration and Installation

To configure, build and install the binary RPM, See [Section 6.2.1 \[Configuring the Binary RPM\]](#), page 24.

6.1.2 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.

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 ‘**streams-source_0.7a.4-0_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 6.3.2 \[Building from the Debian DSC\]](#), page 42).

Architecture Independent

streams-dev_0.7a.4-0-all.deb

The ‘**streams-dev**’ package contains the device definitions necessary to run applications programs developed for Linux Fast-STREAMS.⁶

streams-doc_0.7a.4-0-all.deb

The ‘**streams-doc**’ package contains this manual in plaintext, postscript, PDF and HTML forms, along with the meta-information from the ‘**streams**’ package. It also contains all of the manual pages necessary for developing Linux Fast-STREAMS applications and Linux Fast-STREAMS STREAMS modules or drivers.

⁴ Note that only release packages that contain kernel modules and that export versioned symbols will contain a ‘**info**’ subpackage. Also, this subpackage is only applicable to 2.4 series kernels and is not necessary and not built for 2.6 series kernels.

⁵ Note that the ‘**_kversion**’ of ‘2.4.20-28.7bigmem’ is only an example.

⁶ 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.

streams-init_0.7a.4-0_all.deb

The ‘**streams-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.

streams-source_0.7a.4-0_all.deb

The ‘**streams-source**’ package contains the source code necessary for building the Linux Fast-STREAMS release. It includes the **autoconf** configuration utilities necessary to create and distribute tarballs, rpms and deb/dscs. `!ignore7`
`!end ignore`

Architecture Dependent**streams-devel_0.7a.4-0_i386.deb**

The ‘**streams-devel**’ package contains library archives for static compilation, header files to develop Linux Fast-STREAMS modules and drivers. This also includes the header files and static libraries required to compile Linux Fast-STREAMS applications programs.

streams-lib_0.7a.4-0_i386.deb

The ‘**streams-lib**’ package contains the run-time shared libraries necessary to run application programs and utilities developed for the ‘**streams**’ package.

Kernel-Dependent DEB

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

One of the following Kernel-Dependent packages is required for your architecture and kernel version. If your architecture or kernel version is not on the list, you can build binary DEB from the source DEB (see see [Section 6.3.2 \[Building from the Debian DSC\]](#), page 42).⁸

streams-core-2.4.20-28.7bigmem_0.7a.4-0_i386.deb

The ‘**streams-core**’ package contains the loadable kernel modules that depend only on the kernel. This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version ‘2.4.20-28.7bigmem’.⁹

streams-info-2.4.20-28.7bigmem_0.7a.4-0_i386.deb

The ‘**streams-info**’ package¹⁰ contains the module symbol version information for the ‘**core**’ subpackage, above. It is possible to load this subpackage and

⁷ Note that not all releases have source DEB packages. Release packages that do not contain kernel modules do not generate a source DEB package.

⁸ 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.

⁹ Note that the ‘**_kversion**’ of ‘2.4.20-28.7bigmem’ is only an example. Note also that only release packages that contain kernel modules will contain a ‘**core**’ subpackage.

¹⁰ Note that only release packages that contain kernel modules and that export versioned symbols will contain a ‘**info**’ subpackage. Also, this subpackage is only applicable to 2.4 series kernels and is not necessary and not built for 2.6 series kernels.

compile modules that use the exported symbols without loading the actual kernel modules (from the ‘core’ subpackage above). This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version ‘2.4.20-28.7bigmem’.¹¹

Configuration and Installation

To configure, build and install the Debian DEB, See [Section 6.2.2 \[Configuring the Debian DEB\]](#), page 25.

6.1.3 Downloading the Source RPM

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

`streams-0.7a.4-1.src.rpm`

This is the source RPM for the package. From this source RPM it is possible to build binary RPM for any supported architecture and for any 2.4 or 2.6 kernel.

Configuration

To configure the source RPM, See [Section 6.2.3 \[Configuring the Source RPM\]](#), page 25.

6.1.4 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.

`streams_0.7a.4-0.dsc`

`streams_0.7a.4-0.tar.gz`

This is the Debian DSC for the package. From this Debian DSC it is possible to build binary DEB for any supported architecture and for any 2.4 or 2.6 kernel.

Configuration

To configure the source RPM, See [Section 6.2.4 \[Configuring the Debian DSC\]](#), page 31.

6.1.5 Downloading the Tar Ball

For non-RPM architectures, such as NexusWare embedded target, download the tarball as follows:

`streams-0.7a.4.tar.gz`

`streams-0.7a.4.tar.bz2`

These are the `tar` balls for the release. These `tar` balls contain the `autoconf` 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 architectures and Debian DSC and DEB on DPKG architectures.

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

```
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
```

¹¹ Note that the ‘_kversion’ of ‘2.4.20-28.7bigmem’ is only an example.

or

```
% wget http://www.openss7.org/streams-0.7a.4.tar.gz
```

Unpacking the Archive

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

```
% wget http://www.openss7.org/streams-0.7a.4.tar.gz
% tar -xzvf streams-0.7a.4.tar.gz
```

or

```
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% tar -xjvf streams-0.7a.4.tar.bz2
```

Either will create a subdirectory name ‘streams-0.7a.4’ containing all of the files and subdirectories for the **streams** package.

Configuration

To configure and install the tar ball, See [Section 6.2.5 \[Configuring the Tar Ball\]](#), page 31.

6.1.6 Downloading from CVS

If you are a subscriber or sponsor of **The OpenSS7 Project** with CVS archive access privileges then you can download release or mid-release versions of the ‘streams’ package from the project CVS archive.

The Linux Fast-STREAMS package is located in the ‘streams’ subdirectory of ‘/var/cvs’. For release tag information, see [Chapter 5 \[Releases\]](#), page 13.

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

```
% export CVSROOT='-d:pserver:username@cvs.openss7.com:2401/var/cvs'
% cvs login
Password: *****
% cvs co -r streams_0.7a.4 streams
% cvs logout
```

It is, of course, possible to check out by date or by other criteria. For more information, see [section “cvs\(1\)” in *The Manual Pages*](#).

Preparing the CVS Working Directory

Although public releases of the ‘streams’ 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:

- `autoconf` 2.59
- `automake` 1.9.5
- `libtool` 1.5.14
- `gettext` 0.14.1
- `texinfo` 4.6

It should be stressed that, in particular, the `autoconf` and `automake` must be at version releases 2.59 and 1.9. *The versions normally distributed in mainstream GNU/Linux distributions are, in fact, much older than these versions.*¹² GNU version of these packages configured and installed to default directories will install in `‘/usr/local/’` allowing them to coexist with distribution installed versions.

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

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

```
% autoreconf -fiv streams
```

where, `‘streams’` 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 `‘2.59’`. Otherwise, you may need to perform something like the following:

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

After reconfiguring the directory, the package can then be configured and built using the same instructions as are used for the Tar Ball, see [Section 6.2.5 \[Configuring the Tar Ball\]](#), page 31, and [Section 6.3.3 \[Building from the Tar Ball\]](#), page 43.

Do note, however, that `make` will rebuild the documentation that is normally released with the package. Additional tools may be necessary for building the documentation.

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` and `--enable-dependency-tracking`. The first of these two options will add maintainer-specific targets to any generated `‘Makefile’`, and the later will invoke automatic dependency tracking within the `‘Makefile’` so rebuilds after changes to macro, source or documentation files will be automatically rebuilt.

6.2 Configuration

¹² A notable exception is Debian.

6.2.1 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:

- ‘streams-core-2.4.20-28.7bigmem-0.7a.4-1.7.x.i686.rpm’
 - ‘//lib/modules/2.4.20-28.7bigmem’
 - This relocatable directory contains the kernel modules that provide the streams core, drivers and modules.¹³
- ‘streams-info-2.4.20-28.7bigmem-0.7a.4-1.7.x.i686.rpm’
 - ‘//usr/include/streams/2.4.20-28.7bigmem’
 - This relocatable directory contains the kernel module exported symbol information that allows other kernel modules to be compiled against the correct version of the streams package.¹⁴
- ‘streams-dev-0.7a.4-1.7.x.i686.rpm’
 - (not relocatable)
- ‘streams-devel-0.7a.4-1.7.x.i686.rpm’
 - ‘//usr/lib’
 - This relocatable directory contains streams libraries.
 - ‘//usr/include/streams’
 - This relocatable directory contains streams header files.
- ‘streams-doc-0.7a.4-1.7.x.i686.rpm’
 - ‘//usr/share/doc’
 - This relocatable directory contains all package specific documentation (including this manual). The subdirectory in this directory is the ‘streams-0.7a.4’ directory.
 - ‘//usr/share/info’
 - This relocatable directory contains info files (including the info version of this manual).
 - ‘//usr/share/man’
 - This relocatable directory contains manual pages.
- ‘streams-lib-0.7a.4-1.7.x.i686.rpm’
 - ‘//usr/lib’
 - This relocatable directory contains the run-time shared libraries necessary to run applications programs and utilities developed for Linux Fast-STREAMS.
 - ‘//usr/share/locale’
 - This relocatable directory contains the locale information for shared library files.

¹³ Note that the ‘_kversion’ of ‘2.4.20-28.7bigmem’ is only an example.

¹⁴ Note that the ‘_kversion’ of ‘2.4.20-28.7bigmem’ is only an example. Also, note that the ‘info’ subpackage is only applicable to the 2.4 kernel series.

`'streams-source-0.7a.4-1.7.x.i686.rpm'`

`'//usr/src'`

This relocatable directory contains the source code.

`'streams-util-0.7a.4-1.7.x.i686.rpm'`

`'//usr/bin'`

This relocatable directory contains binary programs and utilities.

`'//usr/sbin'`

This relocatable directory contains system binary programs and utilities.

`'//usr/libexec'`

This relocatable directory contains test programs.

`'//etc'`

This relocatable directory contains init scripts and configuration information.

Installation

To install the binary RPM, See [Section 6.4.1 \[Installing the Binary RPM\]](#), page 46.

6.2.2 Configuring the Debian DEB

In general the binary DEB do not require any configuration.

Installation

To install the Debian DEB, See [Section 6.4.2 \[Installing the Debian DEB\]](#), page 47.

6.2.3 Configuring the Source RPM

When building from the source RPM (see [Section 6.3.1 \[Building from the Source RPM\]](#), page 41), the rebuild process uses a number of macros from the user's `'rpmmacros'` file as described in [section "rpm\(8\)" in *The Manual Pages*](#).

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

```

#
# RPM macros for building rpms
#

%_topdir /usr/src/openss7.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
%_enable_debug_packages 1

#
# Template for debug information sub-package.
# with our little addition of release
#
%debug_package \
%ifnarch noarch\
%global __debug_package 1\
%package debug\
Summary: Debug information for package %{name}\
Group: Development/Debug\
AutoReqProv: 0\
%{?fullrelease:Release: %{fullrelease}}\
%description debug\
This package provides debug information for package %{name}.\
Debug information is useful when developing applications that use this\
package or when debugging this package.\
%files debug -f debugfiles.list\
%defattr(-,root,root)\
%endif\
%{nil}

```

When building from the source RPM (see [Section 6.3.1 \[Building from the Source RPM\]](#), [page 41](#)), it is possible to pass a number of additional configuration options to the `rpmbuild` 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"
```

`--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 `make`. 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` references in manpages.¹⁵ This option will cook `soelim`, `refer`, `tbl` and `pic` commands from the manpages and also strip `groff` comments. The default is to leave manpages uncooked: they are actually smaller that way.

`--with public`

`--without public`

Release public packages or private packages. This option has no effect on the `'streams'` package. The default is to release public packages.

`--with k-debug`

`--without k-debug`

Specifies whether kernel debugging is to be performed on the build kernel modules. Mutually exclusive with `test` and `safe` below. This has the effect of removing static and inline attributes from functions and invoking all debugging macros in the code. The default is to not perform kernel debugging.

¹⁵ In particular, some *Debian* systems do not load the `groff` extensions package and do not have `grefer` installed. Although this is an oversight on the configuration of the particular *Debian* system, we accomodate such misconfiguration with this feature.

`--with k-test`

`--without k-test`

Specifies whether kernel testing is to be performed. Mutually exclusive with `debug` above and `safe` below. This has the effect of removing static and inline attributes from functions and invoking most debugging macros in the code. The default is to not perform kernel testing.

`--with k-safe`

`--without k-safe`

Specifies whether kernel safety 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 place inline. This has the effect of adding the `-finline-functions` flag to `CFLAGS` for compiling kernel modules. Linux 2.4 kernels are normally compiled `-O2` which does not respect the `inline` directive. This compiles kernel modules with `-finline-functions` to get closer to `-O3` optimization. For better optimization controls, See [Section 6.2.5 \[Configuring the Tar Ball\]](#), page 31.

`--with k-modversions`

`--without k-modversions`

Specifies whether kernel symbol versioning is to be applied to symbols exported by package kernel modules. The default is to version exported module symbols. This package does not export symbols so this option has no effect.

`--with devfs`

`--without devfs`

Specifies whether the build is for a device filesystem daemon enabled system with autoloading, or not. The default is to build for `devfsd` autoloading when `CONFIG_DEVFS_FS` is defined in the target kernel. The `rebuild` target uses this option to signal to the RPM spec file that the `'dev'` subpackage need not be built. This option does not appear when the package has no devices.

`--with 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 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 target uses this feature to rebuild for all available architectures and kernels.

In addition, the following rpm options, specific to the Linux Fast-STREAMS package are available:

--with streams-syncqs

When enabled, MP synchronization queues are enabled for SMP kernels. This option defaults to 'disabled'. This option is not tested for early releases.

--without streams-kthreads

When enabled, the STREAMS scheduler runs as a kernel thread. When disabled, the STREAMS scheduler runs as a software interrupt (bottom half). Running the STREAMS scheduler at bottom half instead of a kernel thread breaks the 'strinet' driver, which must be able to invoke kernel functions that might sleep (but don't). This option defaults to 'enabled'.

--without streams-utils

I have experimented with putting the STREAMS utilities into their own package, 'strutil', however, this is not complete yet. This option defaults to 'enabled'. Do not disable this option.

--without big-compile

When enabled, the STREAMS, the Stream head, and the clone driver are all compiled together in one big compilation unit. This allows the compiler greater opportunity to optimize. This option defaults to 'enabled'. Do not disable this option.

--with module-sth

Enable 'sth' (stream head) module linked into 'streams' object. The default is to create the module as a separate loadable kernel module, unless option 'big-compile' is specified. This option defaults to 'disabled'. This option defaults to 'enabled' if 'big-compile' is enabled.

--with module-bufmod

Enable 'bufmod' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.

--with module-nullmod

Enable 'nullmod' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.

--with module-pipemod

Enable 'pipemod' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.

--with module-connld

Enable 'connld' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.

- `--with module-sc`
Enable 'sc' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with module-testmod`
Enable 'testmod' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-clone`
Enable 'clone' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module, unless option 'big-compile' is specified. This option defaults to 'disabled'. This option defaults to 'enabled' if 'big-compile' is enabled.
- `--with driver-echo`
Enable 'echo' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-fifo`
Enable 'fifo' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-log`
Enable 'log' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-loop`
Enable 'loop' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-nsdev`
Enable 'nsdev' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-mux`
Enable 'mux' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-nuls`
Enable 'nuls' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-pipe`
Enable 'pipe' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-sad`
Enable 'sad' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--with driver-sfx`
Enable 'sfx' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.

--with driver-spx

Enable ‘spx’ driver linked into ‘streams’ object. The default is to create the driver as a separate loadable kernel module. This option defaults to ‘disabled’.

--with streams-fifos

Enable override of system fifos with STREAMS-based fifos. This option defaults to ‘disabled’. This option is not tested. Do not enable this option yet.

--without streams-bcm

Disable STREAMS binary compatibility mode. When enabled, exported functions (and callouts) will always pass three arguments in registers on architectures supporting `regparm` (i.e. `__i386__`) regardless of how the kernel was compiled. 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 6.3.1 \[Building from the Source RPM\]](#), page 41.

6.2.4 Configuring the Debian DSC

The Debian DSC can be configured by passing options in the environment variable `BUILD_DEBOPTIONS`. The options placed in this variable take the same form as those passed to the `configure` script, See [Section 6.2.5 \[Configuring the Tar Ball\]](#), page 31. For an example, See [Section 6.3.2 \[Building from the Debian DSC\]](#), page 42.

Build

To build from the Debian DSC, See [Section 6.3.2 \[Building from the Debian DSC\]](#), page 42.

6.2.5 Configuring the Tar Ball

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

6.2.5.1 Configure Options

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

--enable-checks**--disable-checks**

Enable or disable preinstall checks. Each packages supports a number of preinstall checks that can be performed by invoking the ‘`check`’ target with `make`. 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.

--disable-compress-manpages

Compress manpages with ‘gzip -9’ or ‘bzip2 -9’ or leave them uncompressed. The default is to compress manpages with ‘gzip -9’ or ‘bzip2 -9’ if a single compressed manpage exists in the target installation directory (**--mandir**). This disables automatic compression.

--disable-public

Disable public release. Has no effect on the ‘streams’ release. No private components exist in ‘streams’ releases.

--disable-initscripts

Disables the installation of init scripts. The default is to configure and install init scripts and their associated configuration files.

--enable-tools

Specifies whether user space programs and libraries are to be built and installed. The default is to build and install user space programs and libraries. This option can be useful when rebuilding for multiple architectures and target kernels, particularly under rpm. The **rebuild** target uses this feature when rebuilding RPMs for all available architectures and kernels, to rebuild user packages once per architecture instead of once per kernel.

--enable-modules

Specifies whether kernel modules are to be built and installed. The default is to build and install kernel modules. This option can be useful when rebuilding for multiple architectures and target kernels, particularly under rpm. The **rebuild** automake target uses this feature to rebuild for all available architectures and kernels.

--enable-arch

Specifies whether architectural dependent package components are to be built and installed. This option can be useful when rebuilding for multiple architectures and target kernels, particularly under dpkg. The default is to configure, build and install architecture dependent package components.

--enable-indep

Specifies whether architecture independent package components are to be built and installed. This option can be useful when rebuilding for multiple architectures and target kernels, particularly under dpkg. 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** and **--enable-k-debug** below. This has the effect of invoking some more pedantic assertion macros in the code. The default is not to apply kernel safety.

--enable-k-test

Enable kernel module run-time testing. Specifies whether kernel testing is to be performed. This option is mutually exclusive with **--enable-k-safe** above and **--enable-k-debug** below. This has the effect of 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. This has the effect of removing **static** and **inline** attributes from functions and invoking all debug macros in the code (including performance-affecting debug macros). The default is to not perform kernel debugging.

--enable-devfs**--disable-devfs**

Specifies whether the build is for a device filesystem daemon enabled system with autoloading, or not. The default is to build for devfsd autoloading when **CONFIG_DEVFS_FS** is defined in the target kernel. The **rebuild** target uses this option to signal to the RPM spec file that the **'dev'** subpackage need not be built. This option does not appear when the package has no devices.

--with-gpg-user=GNUPGUSER

Specify the **gpg** **'GNUPGUSER'** for signing RPMs and tarballs. The default is the content of the environment variable **GNUPGUSER**. If unspecified, the **gpg** program will normally use the user name of the account invoking the **gpg** 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 nor Debian packages, it applies to the tarball release as a whole. The default is the contents of the **'_pkgepoch'** file in the 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 nor Debian packages, it applies to the tarball release as a whole. The default is the contents of the **'_pkgrelease'** file in the source directory or, if that file does not exist, one (1). This is the number after the last point in the package version number.

--with-pkg-distdir=DIR

Specifies the distribution directory for the package. This is used by the maintainer for building distributions of tarballs. This is the directory into which archives are copied for distribution. The default is the top build directory.

--with-cooked-manpages

Convert manual pages to remove macro dependencies and **grefer** references. Some systems do not like **grefer** references in manpages.¹⁶ This option will cook **soelim**, **refer**, **tbl** and **pic** commands from the manpages and also strip **groff** comments. The default is to leave manpages 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 file '.rpmepoch'.

--with-rpm-release=PACKAGE_RPMRELEASE

Specify the 'PACKAGE_RPMRELEASE' for the RPM rspec file. The default is to use the RPM release contained in the file '.rpmrelease'.

--with-rpm-extra=PACKAGE_RPMEXTRA

Specify the 'PACKAGE_RPMEXTRA' extra release information for the RPM spec file. The default is to use the RPM extra release information contained in the 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 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 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.

¹⁶ In particular, some *Debian* systems do not load the **groff** extensions package and do not have **grefer** installed. Although this is an oversight on the configuration of the particular *Debian* system, we accomodate such misconfiguration with this feature.

`--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-SYMAP-FILE`

Specify the ‘`K-SYMAP-FILE`’ kernel system map file. The default is ‘`K-BUILD-DIR/System.map`’. This file is normally located by the `configure` script and need only be provided for special cross-build environments or when requested by a `configure` script error message.

`--with-k-archdir=K-ARCHDIR`

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

`--with-k-machdir=K-MACHDIR`

Specify the ‘`K-MACHDIR`’ kernel source machine specific directory. The default is ‘`DESTDIR/K-SOURCE-DIR/target_cpu`’. This directory is normally located

by the `configure` script and need only be provided for special cross-build environments or when requested by a `configure` script error message.

`--with-k-config=K-CONFIG`

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

`--with-k-optimize=HOW`

`--without-k-optimize`

Specify 'HOW' optimization, *normal*, *size*, *speed* or *quick*. *size* compiles kernel modules `-Os`, *speed* compiles kernel modules `-O3`, and *quick* compiles kernel modules `-O0`. The default is *normal*. Use with care.

`--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'.

In addition, the following `configure` options, specific to the Linux Fast-STREAMS package are available:

`--enable-streams-syncqs`

When enabled, MP synchronization queues are enabled for SMP kernels. This option defaults to '*disabled*'. This option is not tested for early releases.

`--disable-streams-kthreads`

When enabled, the STREAMS scheduler runs as a kernel thread. When disabled, the STREAMS scheduler runs as a software interrupt (bottom half). Running the STREAMS scheduler at bottom half instead of a kernel thread breaks the '*strinet*' driver, which must be able to invoke kernel functions that might sleep (but don't). This option defaults to '*enabled*'.

`--disable-streams-utils`

I have experimented with putting the STREAMS utilities into their own package, '*strutil*', however, this is not complete yet. This option defaults to '*enabled*'. Do not disable this option.

`--disable-big-compile`

When enabled, the STREAMS, the Stream head, and the clone driver are all compiled together in one big compilation unit. This allows the compiler greater opportunity to optimize. This option defaults to '*enabled*'. Do not disable this option.

`--enable-module-sth`

Enable '*sth*' (stream head) module linked into '*streams*' object. The default is to create the module as a separate loadable kernel module, unless option '*big-compile*' is specified. This option defaults to '*disabled*'. This option defaults to '*enabled*' if '*big-compile*' is enabled.

- `--enable-module-bufmod`
Enable 'bufmod' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-module-nullmod`
Enable 'nullmod' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-module-pipemod`
Enable 'pipemod' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-module-connld`
Enable 'connld' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-module-sc`
Enable 'sc' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-module-testmod`
Enable 'testmod' module linked into 'streams' object. The default is to create the module as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-driver-clone`
Enable 'clone' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module, unless option 'big-compile' is specified. This option defaults to 'disabled'. This option defaults to 'enabled' if 'big-compile' is enabled.
- `--enable-driver-echo`
Enable 'echo' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-driver-fifo`
Enable 'fifo' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-driver-log`
Enable 'log' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.
- `--enable-driver-loop`
Enable 'loop' driver linked into 'streams' object. The default is to create the driver as a separate loadable kernel module. This option defaults to 'disabled'.

--enable-driver-nsdev
 Enable ‘nsdev’ driver linked into ‘streams’ object. The default is to create the driver as a separate loadable kernel module. This option defaults to ‘disabled’.

--enable-driver-mux
 Enable ‘mux’ driver linked into ‘streams’ object. The default is to create the driver as a separate loadable kernel module. This option defaults to ‘disabled’.

--enable-driver-nuls
 Enable ‘nuls’ driver linked into ‘streams’ object. The default is to create the driver as a separate loadable kernel module. This option defaults to ‘disabled’.

--enable-driver-pipe
 Enable ‘pipe’ driver linked into ‘streams’ object. The default is to create the driver as a separate loadable kernel module. This option defaults to ‘disabled’.

--enable-driver-sad
 Enable ‘sad’ driver linked into ‘streams’ object. The default is to create the driver as a separate loadable kernel module. This option defaults to ‘disabled’.

--enable-driver-sfx
 Enable ‘sfx’ driver linked into ‘streams’ object. The default is to create the driver as a separate loadable kernel module. This option defaults to ‘disabled’.

--enable-driver-spx
 Enable ‘spx’ driver linked into ‘streams’ object. The default is to create the driver as a separate loadable kernel module. This option defaults to ‘disabled’.

--enable-streams-fifos
 Enable override of system fifos with STREAMS-based fifos. This option defaults to ‘disabled’. This option is not tested. Do not enable this option yet.

--disable-streams-bcm
 Disable STREAMS binary compatibility mode. When enabled, exported functions (and callouts) will always pass three arguments in registers on architectures supporting `regparm` (i.e. `__i386__`) regardless of how the kernel was compiled. This option defaults to ‘enabled’.

6.2.5.2 Environment Variables

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

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

GNUPGUSER

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

GNUPGHOME

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

GPGPASSWD

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

- SOELIM* Roff source elimination command. This is only necessary when the option `--with-cooked-manpages` has been specified and `configure` cannot find the proper `soelim` command. By default, `configure` will search for this tool.
- REFER* Roff references command. This is only necessary when the option `--with-cooked-manpages` has been specified and `configure` cannot find the proper `refer` command. By default, `configure` will search for this tool.
- TBL* Roff table command. This is only necessary when the option `--with-cooked-manpages` has been specified and `configure` cannot find the proper `tbl` command. By default, `configure` will search for this tool.
- PIC* Roff picture command. This is only necessary when the option `--with-cooked-manpages` has been specified and `configure` cannot find the proper `pic` command. By default, `configure` will search for this tool.
- GZIP* Default compression options provided to `GZIP_CMD`.
- GZIP_CMD*
Manpages (and kernel modules) compression commands. This is only necessary when the option `--without-compressed-manpages` has *not* been specified and `configure` cannot find the proper `gzip` command. By default, `configure` will search for this tool.
- BZIP2* Default compression options provided to `BZIP2_CMD`
- BZIP2_CMD*
Manpages compression commands. This is only necessary when the option `--without-compressed-manpages` has *not* been specified and `configure` cannot find the proper `bzip2` command. By default, `configure` will search for this tool.
- MAKEWHATIS*
Manpages apropros database rebuild command. By default, `configure` will search for this tool. By default, `configure` will search for this tool.
- CHKCONFIG*
Chkconfig command. This was used for installation of init scripts. All packages now come with `init_install` and `init_remove` scripts used to install and remove init scripts on both RPM and debian systems.
- RPM* Rpm command. This is only necessary for RPM builds. By default, `configure` will search for this tool.
- RPMBUILD*
Build RPM command. This is only necessary for RPM builds. By default, `configure` will search for this tool. `rpm` will be used instead of `rpmbuild` only if `rpmbuild` cannot be found.
- DPKG* Dpkg comand. This command is used for building debian packages. By default, `configure` will search for this tool.
- DPKG_SOURCE*
Dpkg-source command. This command is used for building debian dsc packages. By default, `configure` will search for this tool.

DPKG-BUILDPACKAGE

Dpkg-buildpackage command. This command is used for building debian deb packages. By default, **configure** will search for this tool.

DEB-BUILD-ARCH

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

DEB-BUILD-GNU-CPU

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

DEB-BUILD-GNU-SYSTEM

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

DEB-BUILD-GNU-TYPE

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

DEB-HOST-ARCH

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

DEB-HOST-GNU-CPU

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

DEB-HOST-GNU-SYSTEM

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

DEB-HOST-GNU-TYPE

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

LDCONFIG

Configure loader command. 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. For example, for *NexusWare* cross-builds, this is set to environment variable *NEXUSWARE_PREFIX* on configuration to point to the root of the cross-build tree for both configuration and installation.

DEPMOD

Build kernel module dependencies command. 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. 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. 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. 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. This is used for generating module symbol versions during build. By default, **configure** will search for this tool.
- KGENKSYMS**
Linux 2.6 generate kernel symbols command. This is used for generating module symbol version during build. By default, **configure** will search for this tool.
- OBJDUMP**
Object dumping command. This is used for listing information about object files. By default, **configure** will search for this tool.
- NM** Object symbol listing command. This is used for listing information about object files. By default, **configure** will search for this tool.
- MODPOST_CACHE**
Cache file for modpost. 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. 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. This is the executable used by autotest for pre- and post-installation checks. By default, **configure** will search for this tool.

6.2.5.3 Build

To build from the tar ball, See [Section 6.3.3 \[Building from the Tar Ball\]](#), page 43.

6.3 Building

6.3.1 Building from the Source RPM

If you have downloaded the necessary source RPM (see [Section 6.1.3 \[Downloading the Source RPM\]](#), page 21), then the following instructions will rebuild the binary RPMs on your system. Once the binary RPMs are rebuilt, you may install them as described above (see [Section 6.4.1 \[Installing the Binary RPM\]](#), page 46).

The source RPM is rebuilt to binary RPMs as follows:

```
% wget http://www.openss7.org/rpms/SRPMS/streams-0.7a.4-1.src.rpm
% rpmbuild --rebuild -vv streams-0.7a.4-1.src.rpm
```

The rebuild process can also recognize a number of options that can be used to tweak the resulting binaries, See [Section 6.2.3 \[Configuring the Source RPM\]](#), page 25. These options are provided on the `rpm` command line. For example:

```
% rpmbuild --rebuild -vv --target athlon-redhat-linux \
--define "_kversion 2.4.20-28.7bigmem" \
-- streams-0.7a.4-1.src.rpm
```

will rebuild binary RPM for the ‘2.4.20-28.7bigmem’ kernel for the ‘athlon’ architecture.¹⁷

Installation

To install the resulting binary RPM, See [Section 6.4.1 \[Installing the Binary RPM\]](#), page 46.

6.3.2 Building from the Debian DSC

If you have downloaded the necessary Debian DSC (see [Section 6.1.4 \[Downloading the Debian DSC\]](#), page 21), then the following instructions will rebuild the binary DEBs on your system. Once the binary DEBs are rebuilt, you may install them as described above (see [Section 6.4.2 \[Installing the Debian DEB\]](#), page 47).

The Debian DSC is rebuilt to binary DEBs as follows:

```
% wget http://www.openss7.org/debian/streams_0.7a.4-0.dsc
% wget http://www.openss7.org/debian/streams_0.7a.4-0.tar.gz
% dpkg-buildpackage -v streams_0.7a.4-0.dsc
```

The rebuild process can also recognize a number of options that can be used to tweak the resulting binaries, See [Section 6.2.4 \[Configuring the Debian DSC\]](#), page 31. These options are provided in the environment variable `BUILD_DPKG_OPTIONS` and have the same form as the options to `configure`, See [Section 6.2.5 \[Configuring the Tar Ball\]](#), page 31. For example:

```
% BUILD_DEBOPTIONS='
--with-k-release=2.4.20-28.7bigmem
--host=athlon-debian-linux-gnu'
dpkg-buildpackage -v \
streams_0.7a.4-0.dsc
```

will rebuild binary DEB for the ‘2.4.20-28.7bigmem’ kernel for the ‘athlon’ architecture.¹⁸

Installation

To install the resulting binary DEB, See [Section 6.4.2 \[Installing the Debian DEB\]](#), page 47.

¹⁷ Note that the ‘_kversion’ of ‘2.4.20-28.7bigmem’ is only an example.

¹⁸ Note that the ‘_kversion’ of ‘2.4.20-28.7bigmem’ is only an example.

6.3.3 Building from the Tar Ball

If you have downloaded the tar ball (see [Section 6.1.5 \[Downloading the Tar Ball\]](#), page 21), then the following instructions will rebuild the package on your system. (Note that the build process does not require `root` privilege.)

6.3.3.1 Native Build

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

```
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% tar -xjvf streams-0.7a.4.tar.bz2
% pushd streams-0.7a.4
% ./configure
% make
% popd
```

6.3.3.2 Cross-Build

Following is an example for a cross-build. The kernel release version must always be specified for a cross-build.¹⁹ 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` can find them, or they must be specified with declarations such as `'CC=/u5/NexusWare24/ppc-linux/gcc'` on the `configure` command line. Look in the file `'configure.nexusware'` in the release package for an example.

```
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% tar -xjvf streams-0.7a.4.tar.bz2
% pushd streams-0.7a.4
% ./configure DESTDIR="/some/other/root" \
--with-k-release=2.4.18 --host sparc-linux
% make
% popd
```

6.3.3.3 NexusWare Build

Additional support is provided for cross-building for the *Performance Technologies Inc. NexusWare* embedded target for the CPC-384, CPC-388 and CPC-396 cards. A configuration script wrapper (`'configure.nexusware'`) is provided to simplify the cross-build operation for these targets. The following steps describe the process:

1. Follow the normal *NexusWare* instructions for rebuilding a `'generic'` kernel and flash image as follows: (Note that I keep my *NexusWare* build in `'/u5/NexusWare24'`.)

¹⁹ Because it *is* a cross-build, the kernel version on the build machine is unlikely to be the kernel version of the target machine, except by coincidence.

```
% pushd /u5/NexusWare24
% source SETUP.sh
% make
% popd
```

For more recent *NexusWare* releases, the method for rebuilding a kernel is a little different as follows:

```
% pushd /u5/NexusWare80
% ./nexus 2.4
% ./nexus 8260
% ./nexus quick
% . SETUP.sh
% popd
```

2. Next download, unpack (see [Section 6.1.5 \[Downloading the Tar Ball\]](#), page 21) and configure (see [Section 6.2.5 \[Configuring the Tar Ball\]](#), page 31) using the provided ‘`configure.nexusware`’ wrapper for `configure`. This wrapper simply tells the `configure` script where to find the *NexusWare* sources and which *NexusWare* cross-building tools to use for a cross-compile.²⁰

Any of the normal `configure` script options (see [Section 6.2.5 \[Configuring the Tar Ball\]](#), page 31) can be used on the same line as ‘`./configure.nexusware`’. One of particular interest to embedded targets is ‘`--with-k-optimize=size`’ to attempt to reduce the size of the kernel modules.

You must specify the kernel version of the kernel for which you are configuring. Add the `--with-k-release=2.4.18` option for older *NexusWare* releases, `--with-k-release=2.4.25` or `--with-k-release=2.6.12` for more current *NexusWare* releases.

3. Install as normal (see [Section 6.4.3 \[Installing the Tar Ball\]](#), page 47), however, for embedded targets the `install-strip` target should be used instead of the `install` target. The `install-strip` target will strip unnecessary symbols from kernel modules and further reduce the size in the root file system flash image.

Following is what I use for configuration and installation: (My *NexusWare* tree is rooted at ‘`/u5/NexusWare`’.)

²⁰ Although I have not tried it, because we use GNU `autoconf` for configuration, these instructions should work equally well for the Solaris *NexusWare* cross-building environment as it does for the Linux *NexusWare* cross-building environment.

```
% pushd /u5/NexusWare80
% ./nexus 2.4
% ./nexus 8260
% ./nexus quick
% . SETUP.sh
% popd
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% tar -xjvf streams-0.7a.4.tar.bz2
% pushd streams-0.7a.4
% ./configure.nexusware --with-k-release=2.4.25 --with-k-optimize=size
% make
% make DESTDIR="$NEXUSWARE_PREFIX" install-strip
% popd
```

Once built and installed in the *NexusWare* directory, you will have to (currently) hand edit a `.spec` file to include the components you want in the *NexusWare* root file system. If you are cross-building for *NexusWare* you should already know what that means. Objects that you might be interested in copying to the root file system are kernel modules that were installed in `'$NEXUSWARE_PREFIX/lib/modules/2.4.18/streams'`, libraries installed in `'$NEXUSWARE_PREFIX/usr/lib'` and utility functions installed in `'$NEXUSWARE_PREFIX/usr/bin'` and `'$NEXUSWARE_PREFIX/usr/sbin'` and test programs in `'$NEXUSWARE_PREFIX/usr/libexec'`. If you would prefer that these programs be installed in `'$NEXUSWARE_PREFIX/lib'`, `'$NEXUSWARE_PREFIX/bin'`, `'$NEXUSWARE_PREFIX/sbin'` and `'$NEXUSWARE_PREFIX/libexec'`, (say because you want to remote mount the `/usr` directory after boot), then specify the `--exec-prefix=/'` option to `./configure.nexusware`.

In addition, because *NexusWare* does not include an `/etc/modules.conf` file by default, it will be necessary to add one or edit your `rc.4` file to `insmod` the necessary `streams` modules at boot time.

Also, *NexusWare* does not configure its kernels for `CONFIG_KMOD`, so any kernel modules must be loaded by the `rc.4` init script at boot. On more recent *NexusWare* releases, the init scripts will be installed in `'$NEXUSWARE_PREFIX/etc/rc.d/init.d/'` but you must manually edit your `rc.4` script to invoke these scripts.

Once you have completed the necessary `.spec` and `rc.4` file entries, you need to rebuild the `generic` kernel flash image once more for these objects to be included in the flash file system. It is important that this second build of the kernel image be the same as the first.

When modifying and rebuilding a *NexusWare* kernel, it will be necessary to rebuild and install `streams`. Simply perform the last `make install-strip` stage or start again with `./configure.nexusware`. You can place the unpacked tarball in `'$NEXUSWARE_PREFIX/usr/src/streams'`, and add the following to the top-level *NexusWare* `Makefile` to make the build process a single step process instead of dual pass:

```

all:
...
    (cd kernels/generic; $(MAKE) depend)
    (cd usr/src/pcmcia-cs-3.2.1; $(MAKE) config)
    (cd kernels/generic; $(MAKE))
    (cd usr/src/pcmcia-cs-3.2.1; $(MAKE) pti)
    (cd usr/src/pti; $(MAKE))
    (cd drivers; $(MAKE))
    (cd utility; $(MAKE))
#    uncomment for streams build
#    (cd usr/src/streams; ./configure.nexusware; $(MAKE) install-strip)
    (cd build/generic; $(MAKE))
...

```

Another, perhaps simpler approach, is to make the necessary edits to the *NexusWare* top-level ‘Makefile’ and ‘.spec’ and ‘rc.4’ files, download and unpack the tar ball into the *NexusWare* directory, and build the *NexusWare* flash image as normal:

```

% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% pushd /u5/NexusWare24
% source SETUP.sh
% pushd usr/src
% tar -xjvf ${DIRSTACK[2]}/streams-0.7a.4.tar.bz2
% ln -sf streams-0.7a.4 streams
% popd
% make
% popd

```

The situation is a little more complex for recent *NexusWare* releases.

6.4 Installing

6.4.1 Installing the Binary RPM

If you have downloaded the necessary binary RPMs (see [Section 6.1.1 \[Downloading the Binary RPM\]](#), page 17), or have rebuilt binary RPMs using the source RPM (see [Section 6.3.1 \[Building from the Source RPM\]](#), page 41), then the following instructions will install the RPMs on your system. For additional information on `rpm`, see [section “rpm\(8\)” in *The Manual Pages*](#).

```

% pushd RPMS/i686
% rpm -ihv streams-*-0.7a.4-1.7.x.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`, see [section “rpm\(8\)” in **manpages**](#). 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' \
    -- streams-doc-0.7a.4-1.7.x.i686.rpm
```

The previous example will install the ‘streams-doc’ package by will relocate the documentation an info directory contents to the ‘/usr/local’ version.

6.4.2 Installing the Debian DEB

If you have downloaded the necessary Debian DEBs (see [Section 6.1.2 \[Downloading the Debian DEB\]](#), page 19), or have rebuild binary DEBs using the Debian DSC (see [Section 6.3.2 \[Building from the Debian DSC\]](#), page 42), then the following instructions will install the DEBs on your system. For additional information on `dpkg`, see [section “dpkg\(8\)” in *The Manual Pages*](#).

```
% pushd debian
% dpkg -iv streams-*_0.7a.4-0_*.deb
```

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

6.4.3 Installing the Tar Ball

After the build process (see [Section 6.3.3 \[Building from the Tar Ball\]](#), page 43), installation only requires execution of one of two `make` targets:

‘make install’

The ‘install’ `make` target will install all the components of the package. Root privilege is required to successfully invoke this target.

‘make install-strip’

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

6.5 Removing

6.5.1 Removing the Binary RPM

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

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

For more information on `rpm`, see [section “rpm\(8\)” in *The Manual Pages*](#).

6.5.2 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 '^streams-`
```

For more information on `dpkg`, see [section “dpkg\(8\)” in *The Manual Pages*](#).

6.5.3 Removing the Source RPM

To remove all the installed binary RPM build from the source RPM, see [Section 6.5.1 \[Removing the Binary RPM\], page 47](#). Then simply remove the binary RPM package files and source RPM file. A command such as:

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

should remove all ‘streams’ RPMs from your system.

6.5.4 Removing the Debian DSC

To remove all the installed binary DEB build from the Debian DSC, see [Section 6.5.2 \[Removing the Debian DEB\], page 47](#). Then simply remove the binary DEB package files and Debian DSC file. A command such as:

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

should remove all ‘streams’ DEBs, DSCs and TARs from your system.

6.5.5 Removing the Tar Ball

To remove a version installed from tar ball, change to the build directory where the package was built and use the ‘`uninstall`’ make target as follows:

```
% cd /usr/src/streams  
% make uninstall  
% cd ..  
% rm -fr streams-0.7a.4  
% rm -f streams-0.7a.4.tar.gz  
% rm -f streams-0.7a.4.tar.bz2
```

If you have inadvertently removed the build directory and, therefore, no longer have a configured directory from which to execute ‘`make uninstall`’, then perform all of the steps for configuration and installation (see [Section 6.4.3 \[Installing the Tar Ball\], page 47](#)) except the final installation and then perform the steps above.

6.6 Loading

6.6.1 Normal Module Loading

When ‘streams’ installs, modules and drivers are normally configured for demand loading. The ‘install’ and ‘install-strip’ make targets will make the necessary changes to the ‘/etc/modules.conf’ file and place the modules in an appropriate place in ‘//lib/modules/2.4.20-28.7bigmem/streams’. The ‘make install’ process should have copied the kernel module files ‘streams-*.o’ to the directory ‘//lib/modules/2.4.20-28.7bigmem/streams’. This means that to load any of these modules, you can simply execute, for example, ‘modprobe stream-somedriver’.²¹

6.6.1.1 Linux STREAMS Module Loading

The ‘streams’ demand load system supports both the old kerneld and the new kmod mechanisms for demand loading kernel modules.

The convention for ‘streams’ kernel loadable object files is:

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

If your kernel has been built using the ‘kerneld’ daemon, then ‘streams’ 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.streams
if -f /lib/modules/`uname -r`/modules.streams
include /lib/modules/`uname -r`/modules.streams
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.streams’ 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.

6.6.1.2 Linux Fast-STREAMS Module Loading

Linux Fast-STREAMS has a wider range of kernel module loading mechanisms than is provided by *LiS*. For mechanisms used for kernel module loading under *Linux Fast-STREAMS*, See [section “Top” in *Linux Fast-STREAMS Reference Manual*](#).

²¹ Note that the ‘_kversion’ of ‘2.4.20-28.7bigmem’ is only an example.

6.6.2 NexusWare Module Loading

Under exceptional circumstances, such as a *NexusWare* build, it is necessary to hand-edit a `.spec` and `rc.4` file to load the modules at boot time.²²

²² At some time I expect to create an `install-nexusware` target that will make the necessary modifications to the `.spec` and `rc.4` files automatically.

7 Troubleshooting

7.1 Test Suites

7.1.1 Pre-installation Checks

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

When building and installing the package from *RPM* or *DEB* source packages (see [Section 6.3.1 \[Building from the Source RPM\]](#), page 41; and [Section 6.3.2 \[Building from the Debian DSC\]](#), page 42), a fundamental set of post-compile, pre-installation checks are performed prior to building binary packages. This is performed automatically and does not require any special actions on the part of the user creating binary packages from source packages.

When building and installing the package from *tarball* (see [Section 6.3.3 \[Building from the Tar Ball\]](#), page 43; and [Section 6.4.3 \[Installing the Tar Ball\]](#), page 47), however, pre-installation checks are only performed if specifically invoked by the builder of the package. Pre-installation checks are invoked after building the package and before installing the package. Pre-installation checks are performed by invoking the ‘`check`’ or ‘`check.log`’ target to `make` when building the package, as shown in [Example 7.1](#).

```
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% tar -xjvf streams-0.7a.4.tar.bz2
% pushd streams-0.7a.4
% ./configure
% make
% make check # <----- invoke pre-installation checks
% popd
```

Example 7.1: *Invoking Pre-Installation Checks*

Pre-installation checks fall into two categories: *System Checks* and *Maintenance Checks*.

7.1.1.1 Pre-Installation System Checks

System Checks are post-compilation checks that can be performed before installing the package that check to ensure that the compiled objects function and will be successfully installed. When the ‘`--enable-maintainer-mode`’ option has not been passed to `configure`, only *System Checks* will be performed.

For example, the steps shown in [Example 7.2](#) will perform *System* checks.

```
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% tar -xjvf streams-0.7a.4.tar.bz2
% pushd streams-0.7a.4
% ./configure
% make
% make check # <----- invokes System pre-installation checks
% popd
```

Example 7.2: *Invoking System Checks*

7.1.1.2 Pre-Installation Maintenance Checks

Maintenance Checks include all *System Checks*, but also checks to ensure that the kernel modules, applications programs, header files, development tools, test programs, documentation, and manual pages conform to *OpenSS7* standards. When the ‘`--enable-maintainer-mode`’ option has been passed to `configure`, *Maintenance Checks* will be performed.

For example, the steps shown in [Example 7.3](#) will perform *Maintenance* checks.

```
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% tar -xjvf streams-0.7a.4.tar.bz2
% pushd streams-0.7a.4
% ./configure --enable-maintainer-mode
% make
% make check # <----- invokes Maintenance pre-installation checks
% popd
```

Example 7.3: *Invoking Maintenance Checks*

7.1.1.3 Specific Pre-Installation Checks

A number of check scripts are provided in the ‘`scripts`’ subdirectory of the distribution that perform both *System* and *Maintenance* checks. These are as follows:

`check_commands`

This check performs both *System* and *Maintenance* checks.

When performing *System* tests, the following tests are performed:

Unless cross-compiling, or unless a program is included in `AM_INSTALLCHECK_STD_OPTIONS_EXEMPT` every program in `bin_PROGRAMS`, `sbin_PROGRAMS`, and `libexec_PROGRAMS` is tested to ensure that the ‘`--help`’, ‘`--version`’, and ‘`--copying`’ options are accepted. When cross-compiling is not possible to execute cross-compiled binaries, and these checks are skipped in that case.

Script executables, on the other hand, can be executed on the build host, so, unless listed in `AM_INSTALLCHECK_STD_OPTIONS_EXEMPT`, every program in `dist_bit_SCRIPTS`, `dist_sbin_SCRIPTS`, and `pkglibexec_SCRIPTS` are tested to ensure that the ‘`--help`’, ‘`--version`’, and ‘`--copying`’ options are accepted.

When performing *Maintenance* tests, `check_commands` also checks to ensure that a manual page exists in section 1 for every executable binary or script

that will be installed from `bin_PROGRAMS` and `dist_bin_SCRIPTS`. It also checks to ensure that a manual page exists in section 8 for every executable binary or script that will be installed from `sbin_PROGRAMS`, `dist_sbin_SCRIPTS`, `libexec_PROGRAMS`, and `pkglibexec_SCRIPTS`.

`check_decls`

This check only performs *Maintenance* checks.

It collects the results from the `check_libs`, `check_modules` and `check_headers` check scripts and tests to ensure every declaration of a function prototype or external variable contained in installed header files has a corresponding exported symbol from either a to be installed shared object library or a to be installed kernel module. Declarations are exempted from this requirement if their identifiers have been explicitly added to the `EXPOSED_SYMBOL` variable. If `WARN_EXCESS` is set to 'yes', then the check script will only warn when excess declarations exist (without a corresponding exported symbol); otherwise, the check script will generate an error and the check will fail.

`check_headers`

This check only performs *Maintenance* checks.

When performing *Maintenance* tests, it identifies all of the declarations included in to be installed header files. It then checks to ensure that a manual page exists in sections 2, 3, 7 or 9, as appropriate, for the type of declaration. It also checks to see if a manual page source file exists in the source directory for a declaration that has not been included in the distribution. Function or prototype declarations that do not have a manual page in sections 2, 3, or 9 will cause the check to fail. Other declarations (variable, externvar, macro, enumerate, enum, struct, union, typedef, member, etc.) will only warn if a manual page does not exist, but will not fail the check.

`check_libs`

This check only performs *Maintenance* checks.

When performing *Maintenance* tests, it checks that each exported symbol in each to be installed shared object library has a manual page in section 3. It also checks that each exported symbol has a function, prototype or externvar declaration in the to be installed header files. A missing declaration or manual page will cause this check to fail.

`check_mans`

This check only performs *Maintenance* checks.

When performing *Maintenance* tests, it checks that to be install manual pages can be formatted for display without any errors or warnings from the build host `man` program. It also checks that required headings exist for manual pages according to the section in which the manual page will be installed. It warns if recommended headings are not included in the manual pages. Because some *RPM* distributions have manual pages that might conflict with the package manual pages, this check script also checks for conflicts with installed manual pages on the build host. This check script also checks to ensure that all to be

installed manual pages are used in some fashion, that is, they have a declaration, or exported symbol, or are the name of a kernel module or STREAMS module or driver, possibly capitalized.

Note that checking for conflicts with the build host should probably be included in the *System* checks (because *System* checks are performed before the source *RPM %install* scriptlet).

`check_modules`

This check performs both *System* and *Maintenance* checks.

When performing *System* tests, it checks each to be installed kernel module to ensure that all undefined symbols can be resolved to either the kernel or another module. It also checks whether an exported or externally declared symbol conflicts with an exported or externally declared symbol present in the kernel or another module.¹

When performing *Maintenance* tests, this check script tests that each to be installed kernel module has a manual page in section 9 and that each exported symbol that does not begin with an underscore, and that belongs to an exported function or exported variable, has a manual page in section 9. It also checks to ensure that each exported symbol that does not begin with an underscore, and that belongs to an exported function or exported variable, has a function, prototype or externvar declaration in the to be installed header files.

`check_streams`

This check performs only *Maintenance* checks.

When performing *Maintenance* tests, it checks that for each configured *STREAMS* module or driver, or device node, that a manual page exists in section 4 or section 7 as appropriate.

The output of the pre-installation tests are fairly self explanatory. Each check script saves some output to '*name.log*', where *name* is the name of the check script as listed above. A summary of the results of the test are display to standard output and can also be captured to the '*check.log*' file if the '*check.log*' target is used instead of the '*check*' target to *make*.

Because the check scripts proliferate '*name.log*' files throughout the build directory, a '*make check-clean*' *make* target has be provided to clean them out. '*make check-clean*' should be run before each successive run of '*make check*'.

7.1.2 Post-installation Checks

Most OpenSS7 packages ship with a combatibility and conformance test suite built using the '*autotest*' capabilities of '*autoconf 2.59*'. These test suites act as a wrapper for the compatibility and conformance test programs that are shipped with the package.

Unlike the pre-installation checks, the post-installation checks are always run complete. The only check that post-installation test scripts perform is to test whether they have been invoked with root privileges or not. When invoked as root, or as a plain user, some tests might be skipped that require root privileges, or that require plain user privileges, to complete successfully.

¹ This particular check has caught some namespace polution that has occurred in the 2.6.11 kernel.

7.1.2.1 Running Test Suites

There are several ways of invoking the conformance test suites:

1. The test suites can be run after installation of the package by invoking the ‘`make installcheck`’ or ‘`make installcheck.log`’ target. Some packages require that root privileges be acquired before invoking the package.
2. The test suites can be run from the distribution subdirectory after installation of the package by invoking the `testsuite` shell script directly.
3. The test suites can be run standalone from the ‘`libexec`’ (‘`/usr/libexec`’) installation directory by invoking the `testsuite` shell script directly.

Typical steps for invoking the testsuites directly from `make` are shown in [Example 7.4](#).

```
% wget http://www.openss7.org/streams-0.7a.4.tar.bz2
% tar -xjvf streams-0.7a.4.tar.bz2
% pushd streams-0.7a.4
% ./configure
% make
% make check # <----- invokes System pre-installation checks
% make install
% sudo make installcheck # <----- invokes post-installation tests
% popd
```

Example 7.4: *Invoking System Checks*

When performing post-installation checks for the purposes of generating a problem report, the checks should always be performed from the build directory, either with ‘`make installcheck`’ or by invoking `testsuite` directly from the ‘`tests`’ subdirectory of the build directory. This ensures that all of the information known to `configure` and pertinent to the configuration of the system for which a test case failed, will be collected in the resulting ‘`testsuite.log`’ file deposited upon test suite failure in the ‘`tests`’ directory. This ‘`testsuite.log`’ file can then be attached as part of the problem report and provides rich details to maintainers of the package.

7.2 Problem Reports

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'`

A problem occurs with the `'make installcheck'` target that causes the test suite to fail. Attach the resulting `'tests/testsuite.log'` file to the problem report. There is no need to attach the other files as they are included in `'tests/testsuite.log'`.

For other problems that occur during the use of the *Linux Fast-STREAMS* package, please write a test case for the test suite that recreates the problem if one does not yet exist and provide a test program patch with the problem report. Also include whatever log files are generated by the kernel (`cmn_err(9)`) or by the `strerr(8)` or `strace(1)` facilities (`strlog(9)`).

7.3 Known Bugs

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 *Linux Fast-STREAMS* package.

Unverified behaviour may contain unknown bugs.

Please remember that there is **NO WARRANTY**.

Appendix A Copying

A.1 GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

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

A.1.1 Preamble

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

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

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

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

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

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

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

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

A.1.2 Terms and Conditions for Copying, Distribution and Modification

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

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

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

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

3. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
 - c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program,

the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

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

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

4. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

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

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

5. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
6. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore,

by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

7. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
8. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

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

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

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

9. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
10. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

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

version number of this License, you may choose any version ever published by the Free Software Foundation.

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

NO WARRANTY

12. **BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.**
13. **IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.**

END OF TERMS AND CONDITIONS

A.1.3 How to Apply These Terms to Your New Programs

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

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

```
one line to give the program's name and an idea of what it does.
Copyright (C) 19yy name of author
```

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

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

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

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

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

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

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

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

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

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Appendix B Documentation License

B.1 GNU Free Documentation License

GNU FREE DOCUMENTATION LICENSE

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

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

B.1.1 Preamble

The purpose of this License is to make a manual, textbook, or other written document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, 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.

B.1.2 Terms and Conditions for Copying, Distribution and Modification

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The

relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front

cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled “History”, and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled “Acknowledgments” or “Dedications”, preserve the section’s title, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as “Endorsements” or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgments”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that

you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

END OF TERMS AND CONDITIONS

B.1.3 How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being list.
A copy of the license is included in the section entitled ‘‘GNU
Free Documentation License’’.
```

If you have no Invariant Sections, write “with no Invariant Sections” instead of saying which ones are invariant. If you have no Front-Cover Texts, write “no Front-Cover Texts” instead of “Front-Cover Texts being *list*”; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Indices

Index of Concepts

B

binary deps	19, 25, 47, 48
binary rpms	17, 24, 46, 47
bugs	16
bugs, known	56
bugs, reporting	55
building	41
building, source dscs	42
building, source srpm	41
building, tar ball	43

C

checkout, cvs	22
compatibility	9, 13
configuration	23
configure environment variables	38
configure options	31
configuring, binary deps	25
configuring, binary rpms	24
configuring, source dscs	31
configuring, source srpm	25
configuring, tar ball	31
conformance	9
contributors	1
conventions	3
credits	1
cvs	22

D

definitions	3
developing	8
downloading	17
downloading, binary rpms	17
downloading, debian deps	19
downloading, debian dscs	21
downloading, source srpm	21
downloading, tar ball	21
drivers	7

G

GNU/Linux Distributions	13
-------------------------------	----

H

headers	7
history	16

I

installation	17
installing	46

installing, binary deps	47
installing, binary rpms	46
installing, tar ball	47
introduction	3

K

Kernel	14
known bugs	56

L

libraries	8
license, FDL	63
license, GNU Free Documentation License	63
license, GNU General Public License	57
license, GPL	57
licensing	3
Linux STREAMS	14
loading	49
loading kernel modules	50

M

modules	7
---------------	---

N

NexusWare	50
notice	3

O

objective	5
organization	3
overview	3

P

porting	10
post-installation checks	54
pre-installation checks	51
prerequisites	13
problem reports	55

R

reference	7
release notes	14
release streams-0.7a-1	15
release streams-0.7a-2	15
release streams-0.7a-3	15
release streams-0.7a.3	15
release streams-0.7a.4	14
releases	13

removing 47
 removing, binary deps 48
 removing, binary rpms 47
 removing, source dscs 48
 removing, source srpm 48
 removing, tar ball 48
 reporting bugs 55

S

schedule 16
 source dscs 21, 31, 42, 48
 source rpms 21, 25, 41, 48
 sponsors 1
 streams-core-2.4.20-28.7bigmem-0.7a.4-1.7.x.i686-
 .rpm 24
 streams-dev-0.7a.4-1.7.x.i686.rpm 24
 streams-devel-0.7a.4-1.7.x.i686.rpm 24

streams-doc-0.7a.4-1.7.x.i686.rpm 24
 streams-info-2.4.20-28.7bigmem-0.7a.4-1.7.x.i686-
 .rpm 24
 streams-lib-0.7a.4-1.7.x.i686.rpm 24
 streams-source-0.7a.4-1.7.x.i686.rpm 25
 streams-util-0.7a.4-1.7.x.i686.rpm 25

T

tar ball 21, 31, 43, 47, 48
 test suites 51
 test suites, running 55
 troubleshooting 51

U

utilities 8

Index of Data Types

(Index is nonexistent)

Index of Functions and Macros

make.....	49	make install-strip.....	49
make install.....	49		

Index of Variables and Constants

A

AUTOM4TE 41
AUTOTEST 41

B

BZIP2 39
BZIP2_CMD 39

C

CHKCONFIG 39

D

DEB_BUILD_ARCH 40
DEB_BUILD_GNU_CPU 40
DEB_BUILD_GNU_SYSTEM 40
DEB_BUILD_GNU_TYPE 40
DEB_HOST_ARCH 40
DEB_HOST_GNU_CPU 40
DEB_HOST_GNU_SYSTEM 40
DEB_HOST_GNU_TYPE 40
DEPMOD 40
DESTDIR 40
DPKG 39
DPKG_BUILDPACKAGE 40
DPKG_SOURCE 39

G

GENKSYMS 41
GNUPGHOME 38
GNUPGUSER 38
GPG 38
GPGPASSWD 38
GZIP 39
GZIP_CMD 39

K

KGENKSYMS 41

L

LDCONFIG 40
LSMOD 41
LSOF 41

M

MAKEWHATIS 39
MODPOST_CACHE 41
MODPROBE 40

N

NM 41

O

OBJDUMP 41

P

PACKAGE_KVERSION 27
PIC 39

R

REFER 39
RPM 39
RPMBUILD 39

S

SOELIM 39

T

TBL 39

Index of Files and Programs

/	
/lib/modules/2.4.20-28.7bigmem/streams/..	49
S	
specfs.o.....	7
streams-aixcompat.o.....	7
streams-clone.o.....	7
streams-connld.o.....	8
streams-echo.o.....	7
streams-fifo.o.....	7
streams-hpuxcompat.o.....	7
streams-liscompat.o.....	7
streams-log.o.....	7
streams-nsdev.o.....	7
streams-nuls.o.....	7
streams-osfcompat.o.....	7
streams-pipe.o.....	7
streams-pipemod.o.....	8
streams-sad.o.....	7
streams-sc.o.....	8
streams-sth.o.....	8
streams-suncompat.o.....	7
streams-svr4compat.o.....	7
streams-uw7compat.o.....	7
streams.o.....	7

Index of Configuration Options

A

arch 32

B

base-major 36
big-compile 29, 36

C

checks 27, 31
compress-manpages 32
cooked-manpages 27, 34

D

deb-epoch 34
deb-release 34
deb-topdir 34
devfs 33
driver-clone 30, 37
driver-echo 30, 37
driver-fifo 30, 37
driver-log 30, 37
driver-loop 30, 37
driver-mux 30, 38
driver-nsdev 30, 38
driver-nuls 30, 38
driver-pipe 30, 38
driver-sad 30, 38
driver-sfx 30, 38
driver-spx 31, 38

G

gpg-home 33
gpg-user 33

I

indep 32
initscripts 32

K

k-archdir 35
k-build 35
k-config 36

k-debug 27, 33
k-inline 28, 32
k-linkage 35
k-machdir 35
k-modules 35
k-modversions 28
k-optimize 27, 36
k-release 27, 34
k-safe 28, 32
k-sysmap 35
k-test 28, 33

M

module-bufmod 29, 37
module-connld 29, 37
module-nullmod 29, 37
module-pipemod 29, 37
module-sc 30, 37
module-sth 29, 36
module-testmod 30, 37
modules 28, 32

P

pkg-distdir 33
pkg-epoch 33
pkg-release 33
public 27, 32

R

rpm-epoch 34
rpm-extra 34
rpm-release 34
rpm-topdir 34

S

strconf-master 36
streams-bcm 31, 38
streams-fifos 31, 38
streams-kthreads 29, 36
streams-syncqs 29, 36
streams-utils 29, 36

T

tools 28, 32