OpenSS7 ISO Stack
Installation and Reference Manual

Version 0.9.2 Edition 1.rc2
Updated 2006-07-11
Package striso-0.9.2.1.rc2

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

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

This is texinfo edition 1.rc2 of the OpenSS7 ISO Stack documentation, and is consistent with striso 0.9.2. This manual was developed under the OpenSS7 Project and was funded in part by OpenSS7 Corporation.

Short Contents

Table of Contents

Acknowledgments

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

Sponsors

Funding for completion of the OpenSS7 ISO Stack 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 OpenSS7 ISO Stack package is Brian F. G. Bidulock. The following is a list of significant contributors to The OpenSS7 Project:

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

1 Introduction

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

1.1 Notice

This version of `striso` is a version modified by The OpenSS7 Project that contains drivers and modules previously part of *Linux STREAMS*[1]. In stark contrast to many other software packages released by The OpenSS7 Project, this package contains code mostly developed by other parties.

This package is released and distributed under the *GNU General Public License* (see Section A.1 [GNU General Public License], page 69). 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 document is released under the *GNU Free Documentation License* (see Appendix C [FDL], page 85) with all sections invariant.

1.2 Overview

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

The OpenSS7 ISO Stack package is a STREAMS ISO Networking (ISO) package for Linux that can be used with *Linux Fast-STREAMS*[2] or *Linux STREAMS*[3]. It includes development tools, header files and manual pages for

- *Communications Device Interface (CDI)*,
- *Data Link Provider Interface (DLPI)* and
- *Network Provider Interface (NPI)*.

In addition, it provides STREAMS drivers and modules for DLPI including:

- Linux IP to DLPI Driver (‘`streams-ldl.o`’),
- DLPI to Linux IP Driver (‘`streams-ip_to_dlpi.o`’), and
- DLPI to Linux IP Module (‘`streams-ip_strm_mod.o`’).

The OpenSS7 ISO Stack package is essential to the development and support of STREAMS ISO networking modules and drivers and provides a fundamental set of X/Open header files and manual pages for such development.

The OpenSS7 ISO Stack does *not* provide the X/Open Transport Interface (XTI), Transport Provider Interface (TPI) or Transport Layer Interface (TLI) components. See the OpenSS7 `strxnet`[4] and `strinet`[5] packages.

---

[1] See section “About This Manual” in *Linux STREAMS (LiS) Installation and Reference Manual*.
[2] See section “About This Manual” in *Linux Fast-STREAMS (LfS) Reference Manual*.
[3] See section “About This Manual” in *Linux STREAMS (LiS) Reference Manual*.
[4] See section “About This Manual” in *OpenSS7 XTI/TLI Library Reference Manual*.
[5] See section “About This Manual” in *OpenSS7 INET Reference Manual*.

1.3 Organization of this Document

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

1.4 Conventions and Definitions

This manual uses *texinfo* typographic conventions.

2 Reference

2.1 Files

STRISO places the following kernel modules files in the kernel modules directory '`lib/modules/2.4.20-28.7/`':

'`modules.striso`'

STRISO places the following kernel modules files in the kernel modules directory '`lib/modules/2.4.20-28.7/striso/`':

'`streams-ip_strm_mod.o`'
: This kernel module contains the IP to STREAMS module.

'`streams-ip_to_dlpi.o`'
: This kernel module contains the IP to DLPI module.

'`streams-ldl.o`'
: This kernle module contains the Linux DL module.

STRISO places the following header files in the system include directory '`usr/include/striso/`':

'`sys/cdi.h`'
: This file contains the CDI header file.

'`sys/dlpi.h`'
: This file contains the DLPI header file.

'`sys/ldl.h`'
: This file contains the Linux DL header file.

'`sys/npi.h`'
: This file contains the NPI header file.

STRISO places the following test programs in the system libexec directory '`usr/libexec/`':

'`ldltest`' This binary contins a test program for the Linux DL driver.

STRISO places the following utility programs in the system binary directory '`usr/sbin/`':

'`ldlconfig`'
: This binary contains a configuration utility for the Linux DL driver.

'`striso_mknod`'
: This binary conatins a script for making device nodes for the *STRISO* package.

STRISO places the following info files in the system info directory '`usr/share/info/`':

'`striso.info`'
'`striso.info-1`'
'`striso.info-2`'
'`striso.info-3`'
'`striso.info-4`'
: These files contain this manual in info format.

STRISO places the following manpage macro and reference database files in the system man directory '`usr/share/man/`':

'`striso.macros`'
: This file contain manual page macro definitions included by the manual pages included in the package.

'`striso.refs`'
: This file contains a reference database referenced by the manual pages included in the package.

STRISO places the following CDI manual pages in the system man directory '`usr/share/man/man7/`':

'CD_ABORT_OUTPUT_REQ.7', 'CD_ALLOW_INPUT_REQ.7', 'CD_ATTACH_REQ.7', 'CD_BAD_FRAME_IND.7', 'CD_DETACH_REQ.7', 'CD_DISABLE_CON.7', 'CD_DISABLE_REQ.7', 'CD_ENABLE_CON.7', 'CD_ENABLE_REQ.7', 'CD_ERROR_ACK.7', 'CD_ERROR_IND.7', 'CD_HALT_INPUT_REQ.7', 'CD_INFO_ACK.7', 'CD_INFO_REQ.7', 'CD_MODEM_SIG_IND.7', 'CD_MODEM_SIG_POLL.7', 'CD_MODEM_SIG_REQ.7', 'CD_MUX_NAME_REQ.7', 'CD_OK_ACK.7', 'CD_READ_REQ.7', 'CD_UNITDATA_ACK.7', 'CD_UNITDATA_IND.7', 'CD_UNITDATA_REQ.7', 'CD_WRITE_READ_REQ.7', 'cd_abort_output_req_t.7', 'cd_allow_input_req_t.7', 'cd_attach_req_t.7', 'cd_bad_frame_ind_t.7', 'cd_detach_req_t.7', 'cd_disable_con_t.7', 'cd_disable_req_t.7', 'cd_enable_con_t.7', 'cd_enable_req_t.7', 'cd_error_ack_t.7', 'cd_error_ind_t.7', 'cd_halt_input_req_t.7', 'cd_info_ack_t.7', 'cd_modem_sig_ind_t.7', 'cd_modem_sig_poll_t.7', 'cd_modem_sig_req_t.7', 'cd_mux_name_req_t.7', 'cd_ok_ack_t.7', 'cd_read_req_t.7', 'cd_unitdata_ack_t.7', 'cd_unitdata_ind_t.7', 'cd_unitdata_req_t.7', 'cd_write_read_req_t.7', 'cdi.7'

These are CDI manual pages.

STRISO places the following DLPI manual pages in the system man directory '`usr/share/man/man7/`':

'DL_ATTACH_REQ.7', 'DL_BIND_ACK.7', 'DL_BIND_REQ.7', 'DL_CONNECT_CON.7', 'DL_CONNECT_IND.7', 'DL_CONNECT_REQ.7', 'DL_CONNECT_RES.7', 'DL_DATA_ACK_IND.7', 'DL_DATA_ACK_REQ.7', 'DL_DATA_ACK_STATUS_IND.7', 'DL_DATA_IND.7', 'DL_DATA_REQ.7', 'DL_DETACH_REQ.7', 'DL_DISABMULTI_REQ.7', 'DL_DISCONNECT_IND.7', 'DL_DISCONNECT_REQ.7', 'DL_ENABMULTI_REQ.7', 'DL_ERROR_ACK.7', 'DL_GET_STATISTICS_ACK.7', 'DL_GET_STATISTICS_REQ.7', 'DL_INFO_ACK.7', 'DL_INFO_REQ.7', 'DL_OK_ACK.7', 'DL_PHYS_ADDR_ACK.7', 'DL_PHYS_ADDR_REQ.7', 'DL_PROMISCOFF_REQ.7', 'DL_PROMISCON_REQ.7', 'DL_REPLY_IND.7', 'DL_REPLY_REQ.7', 'DL_REPLY_STATUS_IND.7', 'DL_REPLY_UPDATE_REQ.7', 'DL_REPLY_UPDATE_STATUS_IND.7', 'DL_RESET_CON.7', 'DL_RESET_IND.7', 'DL_RESET_REQ.7', 'DL_RESET_RES.7', 'DL_SET_PHYS_ADDR_REQ.7', 'DL_SUBS_BIND_ACK.7', 'DL_SUBS_BIND_REQ.7', 'DL_SUBS_UNBIND_REQ.7', 'DL_TEST_CON.7', 'DL_TEST_IND.7', 'DL_TEST_REQ.7', 'DL_TEST_RES.7', 'DL_TOKEN_ACK.7', 'DL_TOKEN_REQ.7', 'DL_UDERROR_IND.7', 'DL_UDQOS_REQ.7', 'DL_UNBIND_REQ.7', 'DL_UNITDATA_IND.7', 'DL_UNITDATA_REQ.7', 'DL_XID_CON.7', 'DL_XID_IND.7', 'DL_XID_REQ.7', 'DL_XID_RES.7', 'dl_attach_req_t.7', 'dl_bind_ack_t.7', 'dl_bind_req_t.7', 'dl_connect_con_t.7', 'dl_connect_ind_t.7', 'dl_connect_req_t.7', 'dl_connect_res_t.7', 'dl_data_ack_ind_t.7', 'dl_data_ack_req_t.7', 'dl_data_ack_status_ind_t.7', 'dl_detach_req_t.7',

‘dl_disabmulti_req_t.7’, ‘dl_disconnect_ind_t.7’, ‘dl_disconnect_req_t.7’, ‘dl_enabmulti_req_t.7’, ‘dl_error_ack_t.7’, ‘dl_get_statistics_ack_t.7’, ‘dl_get_statistics_req_t.7’, ‘dl_info_ack_t.7’, ‘dl_info_req_t.7’, ‘dl_ok_ack_t.7’, ‘dl_phys_addr_ack_t.7’, ‘dl_phys_addr_req_t.7’, ‘dl_promiscoff_req_t.7’, ‘dl_promiscon_req_t.7’, ‘dl_reply_ind_t.7’, ‘dl_reply_req_t.7’, ‘dl_reply_status_ind_t.7’, ‘dl_reply_update_req_t.7’, ‘dl_reply_update_status_ind_t.7’, ‘dl_reset_con_t.7’, ‘dl_reset_ind_t.7’, ‘dl_reset_req_t.7’, ‘dl_reset_res_t.7’, ‘dl_set_phys_addr_req_t.7’, ‘dl_subs_bind_ack_t.7’, ‘dl_subs_bind_req_t.7’, ‘dl_subs_unbind_req_t.7’, ‘dl_test_con_t.7’, ‘dl_test_ind_t.7’, ‘dl_test_req_t.7’, ‘dl_test_res_t.7’, ‘dl_token_ack_t.7’, ‘dl_token_req_t.7’, ‘dl_uderror_ind_t.7’, ‘dl_udqos_req_t.7’, ‘dl_unbind_req_t.7’, ‘dl_unitdata_ind_t.7’, ‘dl_unitdata_req_t.7’, ‘dl_xid_con_t.7’, ‘dl_xid_ind_t.7’, ‘dl_xid_req_t.7’, ‘dl_xid_res_t.7’, ‘dlpi.7’, ‘dlpi_eth.7’

These are DLPI manual pages.

STRISO places the following NPI manual pages in the system man directory ‘usr/share/man/man7/’:

‘N_BIND_ACK.7’, ‘N_BIND_REQ.7’, ‘N_CONN_CON.7’, ‘N_CONN_IND.7’, ‘N_CONN_REQ.7’, ‘N_CONN_RES.7’, ‘N_DATACK_IND.7’, ‘N_DATACK_REQ.7’, ‘N_DATA_IND.7’, ‘N_DATA_REQ.7’, ‘N_DISCON_IND.7’, ‘N_DISCON_REQ.7’, ‘N_ERROR_ACK.7’, ‘N_EXDATA_IND.7’, ‘N_EXDATA_REQ.7’, ‘N_INFO_ACK.7’, ‘N_INFO_REQ.7’, ‘N_OK_ACK.7’, ‘N_OPTMGMT_REQ.7’, ‘N_RESET_CON.7’, ‘N_RESET_IND.7’, ‘N_RESET_REQ.7’, ‘N_RESET_RES.7’, ‘N_TOKEN_ACK.7’, ‘N_TOKEN_REQ.7’, ‘N_UDERROR_IND.7’, ‘N_UNBIND_REQ.7’, ‘N_UNITDATA_IND.7’, ‘N_UNITDATA_REQ.7’ ‘N_bind_ack_t.7’, ‘N_bind_req_t.7’, ‘N_conn_con_t.7’, ‘N_conn_ind_t.7’, ‘N_conn_req_t.7’, ‘N_conn_res_t.7’, ‘N_data_ind_t.7’, ‘N_data_req_t.7’, ‘N_datack_ind_t.7’, ‘N_datack_req_t.7’, ‘N_discon_ind_t.7’, ‘N_discon_req_t.7’, ‘N_error_ack_t.7’, ‘N_exdata_ind_t.7’, ‘N_exdata_req_t.7’, ‘N_info_ack_t.7’, ‘N_info_req_t.7’, ‘N_ok_ack_t.7’, ‘N_optmgmt_req_t.7’, ‘N_reset_con_t.7’, ‘N_reset_ind_t.7’, ‘N_reset_req_t.7’, ‘N_reset_res_t.7’, ‘N_token_ack_t.7’, ‘N_token_req_t.7’, ‘N_uderror_ind_t.7’, ‘N_unbind_req_t.7’, ‘N_unitdata_ind_t.7’, ‘N_unitdata_req_t.7’, ‘npi.7’

These are NPI manual pages.

2.2 Drivers

2.2.1 Linux IP to DLPI Driver (ldl)

The LDL driver sits atop any existing Linux IP interface driver and presents a DLPI interface to the STREAMS drivers above. This allows any STREAMS driver that communicates downstream using DLPI to utilize the services of existing Linux drivers for Ethernet, Token Ring, etc.

Licensing

The LDL driver and the `ldltest` and `ldlconfig` programs were originally written by Ole Husgaard and is licensed under the GNU General Public License, See Section A.1 [GNU

General Public License], page 69. This OpenSS7 release of the LDL driver also includes Hewlett-Packard proposed patches. An OpenSS7 GPL header has been added to the source code files to make this clear.

2.2.2 DLPI to Linux IP Driver (ip_to_dlpi)

Licensing

The ip_to_dlpi driver was originally written by *The Software Group Limited* and in licensed under the GNU Lesser General Public License See Appendix B [LGPL], page 75. Nevertheless, this OpenSS7 release of the '`ip_to_dlpi`' driver is released under the GNU General Public License See Section A.1 [GNU General Public License], page 69. An OpenSS7 GPL header has been added to the source code to make this clear.

2.3 Modules

2.3.1 DLPI to Linux IP Module (ip_strm_mod)

Licensing

The '`ip_strm_mod`' module was originally written by Mikel L. Mathews and is licensed under the GNU Lesser General Public License See Appendix B [LGPL], page 75. However, this OpenSS7 release is distributed under the GNU General Public License See Section A.1 [GNU General Public License], page 69. An OpenSS7 GPL header has been added to the source code to make this clear.

LiS includes two adapter drivers to assist in interfacing STREAMS drivers to the Linux Kernel's TCP/IP protocols. One driver, *ip_strm_mod* acts as an IP interface driver. It fits below IP using standard `ifconfig`[1] procedures. It, in turn, communicates downstream with any STREAMS driver using the DLPI protocol in a fashion similar to the manner in which IP on Unix systems interfaces to lower interface drivers. This allows a DLPI STREAMS driver to act as an interface driver to Linux TCP/IP.

2.4 Utilities

2.4.1 `ldltest`

Note that `ldltest` is mainatined as a manual page, section "ldltest(8)" in *The Manual Pages*.

2.4.2 `ldlconfig`

Note that `ldlconfig` is mainatined as a manual page, section "ldlconfig(8)" in *The Manual Pages*.

2.5 Development

[1] See section "ifconfig(8)" in *The Manual Pages*.

3 Conformance

4 Releases

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

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

The following sections provide information on OpenSS7 ISO Stack releases as well as compatibility information of OpenSS7 release to the original GCOM releases of these modules and drivers, as well as Linux kernel compatibility.

4.1 Prerequisites

Prerequisites for the OpenSS7 ISO Stack package are as follows:

- A fairly LSB compliant GNU/Linux distribution.[4]
- Linux 2.4 or 2.6 kernel (2.4.10 - 2.4.27) or (2.6.3 - 2.6.15)
- glibc2 or better.
- GNU info (for info files).
- GNU groff (for man pages).[5]
- OpenSS7 Linux Fast-STREAMS (streams-0.7a.5 or better)[6]
- OpenSS7 STREAMS Compatibility Modules (strcompat-0.9.2.3 or better)

If you need to rebuild the package from sources with modifications, you will need a larger GNU toolchain as described in See Section 5.1.6 [Downloading from CVS], page 25.

4.2 Compatibility

This section discusses compatibility with major prerequisites.

4.2.1 GNU/Linux Distributions

OpenSS7 ISO Stack is compatible with the following *Linux* distributions:[7]

- RedHat Linux 7.2 (RH7)
- RedHat Linux 7.3 (RH7)
- Performance Technlogies *NexusWare24* – TBD

---

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

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

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

[4] See Section 4.2.1 [GNU/Linux Distributions], page 11, for more information.

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

[6] Although this an other OpenSS7 STREAMS packages will build and install with Linux STREAMS (LiS), **Linux STREAMS (LiS) is deprecated and cannot be supported because it is full of bugs!**

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

- RedHat Linux 8.0 (RH8) – TBD
- RedHat Linux 9 (RH9) – TBD
- SuSE 8.0 Professional – TBD
- Fedora Core 1 (FC1) – TBD
- Debian 3.0r2 Woody
- Mandrakelinux 9.2 (MDK92) – TBD
- RedHat Enterprise Linux 3.0 (EL3)
- WhiteBox Enterprise Linux 3.0 (WBEL3)
- CentOS Enterprise Linux 3.4 (centos34)
- Fedora Core 2 (FC2) – TBD
- SuSE 9.1 Personal – TBD
- Mandrakelinux 10.0 (MDK100) – TBD
- SuSE 9.2 Professional (SuSE9.2) – TBD
- Mandrakelinux 10.1 (MDK101) – TBD
- Fedora Core 3 (FC3) – TBD
- RedHat Enterprise Linux 4 (EL4)
- CentOS Enterprise Linux 4.0 (centos4)
- WhiteBox Enterprise Linux 4 (WBEL4)
- Fedora Core 4 (FC4)
- Lineox 4.026 (LEL4) – TBD
- Lineox 4.053 (LEL4)
- Mandriva Linux LE2005 (MDK102) – TBD
- Performance Technologies NexusWare 8.0
- Debian 3.1r0a Sarge (untested)
- SuSE 10.0 (untested)
- OpenSuSE (untested)
- Mandriva Linux LE2006 (MDK103) (untested)

When installing from the tarball (see Section 5.4.3 [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.

4.2.2 Kernel

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

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

`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 *OpenSS7 ISO Stack* package is compatible with 2.4 kernel series after 2.4.10 and has been tested up to and including 2.4.27. It has been tested from 2.6.3 up to and including 2.6.15.

UP validation testing for kernels is performed on all supported architectures. SMP validation testing is performed on UP machines, as well as on an Intel 3.0GHz Pentium IV 630 with HyperThreading enabled. Because HyperThreading is not as independent as multiple CPUs, SMP validation testing is limited.

4.2.3 Architectures

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

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

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

4.2.4 Linux STREAMS

The *OpenSS7 ISO Stack* package is currently compatible with *Linux STREAMS*,[9] however, to use the *OpenSS7 ISO Stack* package with *LiS* requires use of the OpenSS7 release packages of *LiS*. The *OpenSS7 ISO Stack* package is compatible with the OpenSS7 **LiS-2.18.3** release that is available from the The OpenSS7 Project Downloads Page. But, do not use *LiS*: it is buggy, unsupported and deprecated. Use *Linux Fast-STREAMS* instead.

4.2.5 Linux Fast-STREAMS

The *OpenSS7 ISO Stack* package is currently compatible with *Linux Fast-STREAMS (LfS)*. The *OpenSS7 ISO Stack* package is compatible with the OpenSS7 **streams-0.7a.4** release that is available from the The OpenSS7 Project Downloads Page.

4.3 Release Notes

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

4.3.1 Release striso-0.9.2.1.rc2

Initial autoconf/RPM packaging of the `striso` release.

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

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

4.4 Maturity

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

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

4.4.1 Pre-Alpha Releases

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

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

4.4.2 Alpha Releases

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

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

4.4.3 Beta Releases

Beta releases are releases that have received some testing, but the testing to date is not exhaustive. *Beta* release packages do not ship with known defects. All known defects are respolved before distribution; however, as exhaustive testing has not been performed, unknown defects may exist. The purpose for a *beta* release is to provide a baseline for other organizations to participate in the rigorous testing of the package.

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

4.4.4 Gamma Releases

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

Gamma release packages typically resolve problems in previous *beta* releases, and might not have had full regression testing performed. Suitability of software in this category for production use is at the discretion of the user of the software. *The OpenSS7 Project*

recommends that the complete validation test suites provided with the package be performed and pass on target systems before considering production use.

4.4.5 Production Releases

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

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

4.5 Bugs

4.5.1 Defect Notices

OpenSS7 ISO Stack has unknown defects. This is a *alpha* release. Some defects could be harmful. No validation testing whatsoever has been performed by the *OpenSS7 Project* on this software. The software might fail to configure or compile on some systems. The *OpenSS7 Project* recommends that you **do not use this software for purposes other than develoment or evaluation, and then only with great care**. Use at your own risk. Remember that there is **NO WARRANTY**.[10]

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

4.5.2 Known Defects

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

4.6 Schedule

4.7 History

[10] See section **NO WARRANTY** under Section A.1 [GNU General Public License], page 69.

5 Installation

5.1 Downloading

The OpenSS7 ISO Stack 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 `striso` from the links in the sections that follow.

By far the easiest form for installing and using `striso-0.9.2.1.rc2` 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 `striso-0.9.2.1.rc2` 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.

5.1.1 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, as applicable. Some release packages do not provide RPMs in each of the several categories.

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

Independent RPM

Independent RPM are dependent on neither the Linux kernel version, nor the STREAMS package. For example, the source package '`striso-source-0.9.2.1.rc2-1.FC5.noarch.rpm`', is not dependent on kernel nor STREAMS package.

All of the following kernel and STREAMS independent RPM are required for your architecture. Binary RPMs listed here are for example only: additional binary RPMs are available from the downloads site. If your architecture is not available, you can build binary RPM from the source RPM (see see Section 5.3.1 [Building from the Source RPM], page 41).

Architecture Independent

striso-dev-0.9.2.1.rc2-1.FC5.noarch.rpm
: The '`striso-dev`' package contains the device definitions necessary to run applications programs developed for OpenSS7 ISO Stack.[1]

striso-doc-0.9.2.1.rc2-1.FC5.noarch.rpm
: The '`striso-doc`' package contains this manual in plaintext, postscript, PDF and HTML forms, along with the meta-information from the '`striso`' package.

[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.

It also contains all of the manual pages necessary for developing OpenSS7 ISO Stack applications and OpenSS7 ISO Stack STREAMS modules or drivers.

striso-init-0.9.2.1.rc2-1.FC5.noarch.rpm
: The '`striso-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.

striso-source-0.9.2.1.rc2-1.FC5.noarch.rpm
: The '`striso-source`' package contains the source code necessary for building the OpenSS7 ISO Stack release. It includes the `autoconf` configuration utilities necessary to create and distribute tarballs, rpms and deb/dscs.[2]

Architecture Dependent

striso-devel-0.9.2.1.rc2-1.FC5.i686.rpm
: The '`striso-devel`' package contains library archives for static compilation, header files to develop OpenSS7 ISO Stack modules and drivers. This also includes the header files and static libraries required to compile OpenSS7 ISO Stack applications programs.

striso-lib-0.9.2.1.rc2-1.FC5.i686.rpm
: The '`striso-lib`' package contains the run-time shared libraries necessary to run application programs and utilities developed for the '`striso`' package.[3]

STREAMS-Dependent RPM

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

One of the following STREAMS-Dependent packages is required for your architecture. If your architecture is not on the list, you can build binary RPM from the source RPM (see see Section 5.3.1 [Building from the Source RPM], page 41).

striso-LiS-util-0.9.2.1.rc2-1.FC5.i686.rpm
: The '`striso-LiS-util`' package provides administrative and configuration test utilities and commands associated with the OpenSS7 ISO Stack package. Because this package must link a STREAMS-specific library, it is a STREAMS-Dependent package. Use the '`striso-LiS-util`' package if you have LiS installed.

striso-streams-util-0.9.2.1.rc2-1.FC5.i686.rpm
: The '`striso-streams-util`' package provides administrative and configuration test utilities and commands associated with the OpenSS7 ISO Stack package. Because this package must link a STREAMS-specific library, it is a

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

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

STREAMS-Dependent package. Use the '`striso-streams-util`' package if you have streams installed.

Kernel-Dependent RPM

Kernel-Dependent RPM are dependent on specific Linux Kernel Binary RPM releases. Packages are provided for popular released *RedHat* kernels. Packages dependent upon *RedHat* 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 5.3.1 [Building from the Source RPM], page 41).[4]

striso-core-2.6.17-1.2139˙FC5-0.9.2.1.rc2-1.FC5.i686.rpm

The '`striso-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 '`2.6.17-1.2139_FC5`'.[5]

striso-info-2.6.17-1.2139˙FC5-0.9.2.1.rc2-1.FC5.i686.rpm

The '`striso-info`' package[6] contains the module symbol version information for the '`core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loading the actual kernel modules (from the '`core`' subpackage above). This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version '`2.6.17-1.2139_FC5`'.[7]

striso-LiS-core-2.6.17-1.2139˙FC5-0.9.2.1.rc2-1.FC5.i686.rpm

The '`striso-LiS-core`' package contains the kernel modules that provide the OpenSS7 ISO Stack STREAMS modules and drivers. This package is heavily tied to the STREAMS package and kernel for which it was compiled. This particular package applies to '`LiS`' (*Linux STREAMS*) on kernel version '`2.6.17-1.2139_FC5`'.[8]

striso-streams-core-2.6.17-1.2139˙FC5-0.9.2.1.rc2-1.FC5.i686.rpm

The '`striso-streams-core`' package contains the kernel modules that provide the OpenSS7 ISO Stack STREAMS modules and drivers. This package is heavily tied to the STREAMS package and kernel for which it was compiled. This particular package applies to '`streams`' (*Linux Fast-STREAMS*) on kernel version '`2.6.17-1.2139_FC5`'.[9]

---

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

[5] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example. Note also that only release packages that contain kernel modules will contain a '`core`' subpackage.

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

[7] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

[8] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

[9] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

striso-LiS-info-2.6.17-1.2139˙FC5-0.9.2.1.rc2-1.FC5.i686.rpm
: The '`striso-LiS-info`' package[10] contains the module symbol version information for the '`LiS-core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loaded the actual kernel modules (from the '`LiS-core`' subpackage above). This package is heavily tied to the STREAMS package and kernel for which it was compiled. This particular package applies to '`LiS`' (*Linux STREAMS*) on kernel version '`2.6.17-1.2139_FC5`'.[11]

striso-streams-info-2.6.17-1.2139˙FC5-0.9.2.1.rc2-1.FC5.i686.rpm
: The '`striso-streams-info`' package[12] contains the module symbol version information for the '`streams-core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loaded the actual kernel modules (from the '`streams-core`' subpackage above). This package is heavily tied to the STREAMS package and kernel for which it was compiled. This particular package applies to '`streams`' (*Linux Fast-STREAMS*) on kernel version '`2.6.17-1.2139_FC5`'.[13]

Configuration and Installation

To configure, build and install the binary RPM, See Section 5.2.1 [Configuring the Binary RPM], page 27.

5.1.2 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, as applicable. Some release packages do not provide DEBs in each of the several categories.

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

Independent DEB

Independent DEB are dependent on neither the Linux kernel version, nor the STREAMS package. For example, the source package '`striso-source_0.9.2.1.rc2-0_i386.deb`', is not dependent on kernel nor STREAMS package.

All of the following kernel and STREAMS independent DEB are required for your architecture. Binary DEBs listed here are for example only: additional binary DEBs are available from the downloads site. If your architecture is not available, you can build binary DEB from the Debian DSC (see see Section 5.3.2 [Building from the Debian DSC], page 42).

---

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

[11] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

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

[13] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

Architecture Independent

striso-dev_0.9.2.1.rc2-0_all.deb
: The '`striso-dev`' package contains the device definitions necessary to run applications programs developed for OpenSS7 ISO Stack.[14]

striso-doc_0.9.2.1.rc2-0_all.deb
: The '`striso-doc`' package contains this manual in plaintext, postscript, PDF and HTML forms, along with the meta-information from the '`striso`' package. It also contains all of the manual pages necessary for developing OpenSS7 ISO Stack applications and OpenSS7 ISO Stack STREAMS modules or drivers.

striso-init_0.9.2.1.rc2-0_all.deb
: The '`striso-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.

striso-source_0.9.2.1.rc2-0_all.deb
: The '`striso-source`' package contains the source code necessary for building the OpenSS7 ISO Stack release. It includes the `autoconf` configuration utilities necessary to create and distribute tarballs, rpms and deb/dscs.[15]

Architecture Dependent

striso-devel_0.9.2.1.rc2-0_i386.deb
: The '`striso-devel`' package contains library archives for static compilation, header files to develop OpenSS7 ISO Stack modules and drivers. This also includes the header files and static libraries required to compile OpenSS7 ISO Stack applications programs.

striso-lib_0.9.2.1.rc2-0_i386.deb
: The '`striso-lib`' package contains the run-time shared libraries necessary to run application programs and utilities developed for the '`striso`' package.[16]

STREAMS-Dependent DEB

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

One of the following STREAMS-Dependent packages is required for your architecture. If your architecture is not on the list, you can build binary DEB from the Debian DSC (see see Section 5.3.2 [Building from the Debian DSC], page 42).

---

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

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

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

striso-LiS-util_0.9.2.1.rc2-0_i386.deb
: The '`striso-LiS-util`' package provides administrative and configuration test utilities and commands associated with the OpenSS7 ISO Stack package. Because this package must link a STREAMS-specific library, it is a STREAMS-Dependent package. Use the '`striso-LiS-util`' package if you have LiS installed.

striso-streams-util_0.9.2.1.rc2-0_i386.deb
: The '`striso-streams-util`' package provides administrative and configuration test utilities and commands associated with the OpenSS7 ISO Stack package. Because this package must link a STREAMS-specific library, it is a STREAMS-Dependent package. Use the '`striso-streams-util`' package if you have streams installed.

Kernel-Dependent DEB

Kernel-Dependent DEB are dependent on specific Linux Kernel Binary DEB releases. Packages are provided for popular released *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 5.3.2 [Building from the Debian DSC], page 42).[17]

striso-core-2.6.17-1.2139˙FC5_0.9.2.1.rc2-0_i386.deb
: The '`striso-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 '`2.6.17-1.2139_FC5`'.[18]

striso-info-2.6.17-1.2139˙FC5_0.9.2.1.rc2-0_i386.deb
: The '`striso-info`' package[19] contains the module symbol version information for the '`core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loading the actual kernel modules (from the '`core`' subpackage above). This package is heavily tied to the kernel for which it was compiled. This particular package applies to kernel version '`2.6.17-1.2139_FC5`'.[20]

striso-LiS-core-2.6.17-1.2139˙FC5_0.9.2.1.rc2-0_i386.deb
: The '`striso-LiS-core`' package contains the kernel modules that provide the OpenSS7 ISO Stack STREAMS modules and drivers. This package is heavily tied to the STREAMS package and kernel for which it was compiled.

---

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

[18] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example. Note also that only release packages that contain kernel modules will contain a '`core`' subpackage.

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

[20] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

This particular package applies to '`LiS`' (*Linux STREAMS*) on kernel version '`2.6.17-1.2139_FC5`'.[21]

striso-streams-core-2.6.17-1.2139˙FC5_0.9.2.1.rc2-0_i386.deb
: The '`striso-streams-core`' package contains the kernel modules that provide the OpenSS7 ISO Stack STREAMS modules and drivers. This package is heavily tied to the STREAMS package and kernel for which it was compiled. This particular package applies to '`streams`' (*Linux Fast-STREAMS*) on kernel version '`2.6.17-1.2139_FC5`'.[22]

striso-LiS-info-2.6.17-1.2139˙FC5_0.9.2.1.rc2-0_i386.deb
: The '`striso-LiS-info`' package[23] contains the module symbol version information for the '`LiS-core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loaded the actual kernel modules (from the '`LiS-core`' subpackage above). This package is heavily tied to the STREAMS package and kernel for which it was compiled. This particular package applies to '`LiS`' (*Linux STREAMS*) on kernel version '`2.6.17-1.2139_FC5`'.[24]

striso-streams-info-2.6.17-1.2139˙FC5_0.9.2.1.rc2-0_i386.deb
: The '`striso-streams-info`' package[25] contains the module symbol version information for the '`streams-core`' subpackage, above. It is possible to load this subpackage and compile modules that use the exported symbols without loaded the actual kernel modules (from the '`streams-core`' subpackage above). This package is heavily tied to the STREAMS package and kernel for which it was compiled. This particular package applies to '`streams`' (*Linux Fast-STREAMS*) on kernel version '`2.6.17-1.2139_FC5`'.[26]

Configuration and Installation

To configure, build and install the Debian DEB, See Section 5.2.2 [Configuring the Debian DEB], page 28.

5.1.3 Downloading the Source RPM

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

striso-0.9.2.1.rc2-1.src.rpm
: This is the source RPM for the package. From this source RPM it is possible to build binary RPM for any supported architecture and for any 2.4 or 2.6 kernel, for either *Linux STREAMS* or *Linux Fast-STREAMS*.

---

[21] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

[22] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

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

[24] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

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

[26] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

Configuration

To configure the source RPM, See Section 5.2.3 [Configuring the Source RPM], page 28.

5.1.4 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.

striso_0.9.2.1.rc2-0.dsc
striso_0.9.2.1.rc2-0.tar.gz
: This is the Debian DSC for the package. From this Debian DSC it is possible to build binary DEB for any supported architecture and for any 2.4 or 2.6 kernel, for either *Linux STREAMS* or *Linux Fast-STREAMS*.

Configuration

To configure the source RPM, See Section 5.2.4 [Configuring the Debian DSC], page 32.

5.1.5 Downloading the Tar Ball

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

striso-0.9.2.1.rc2.tar.gz
striso-0.9.2.1.rc2.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/striso-0.9.2.1.rc2.tar.bz2
```

or

```
% wget http://www.openss7.org/striso-0.9.2.1.rc2.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/striso-0.9.2.1.rc2.tar.gz
% tar -xzvf striso-0.9.2.1.rc2.tar.gz
```

or

```
% wget http://www.openss7.org/striso-0.9.2.1.rc2.tar.bz2
% tar -xjvf striso-0.9.2.1.rc2.tar.bz2
```

Either will create a subdirectory name '`striso-0.9.2.1.rc2`' containing all of the files and subdirectories for the `striso` package.

Configuration

To configure and install the tar ball, See Section 5.2.5 [Configuring the Tar Ball], page 32.

5.1.6 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 '`striso`' package from the project CVS archive.

The OpenSS7 ISO Stack package is located in the '`striso`' subdirectory of '`/var/cvs`'. For release tag information, see Chapter 4 [Releases], page 11.

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 striso_0.9.2.1.rc2 striso
% 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 '`striso`' 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 1.9.6`
- `libtool 1.5.22`
- `gettext 0.14.5`

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.*[27] GNU version of these packages configured and installed to default directories will install in '`/usr/local/`' allowing them to coexist with distribution installed versions.

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

- `gs 8.15`
- `tetex 3.0`
- `texinfo 4.8`

[27] A notable exception is Debian.

- `transfig 3.2.5`
- `imagemagick 6.2.4`
- `groff 1.17.2`

Most desktop GNU/Linux distributions will have these tools; however, some server-style installations (e.g. *Ubuntu*-server) will not and they must be installed separately.

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

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

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

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

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

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

```
% autoreconf -fiv striso
```

where, '`striso`' 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 striso
```

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 5.2.5 [Configuring the Tar Ball], page 32, and Section 5.3.3 [Building from the Tar Ball], page 42.

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 advan-

tage 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.

5.2 Configuration

5.2.1 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:

'`striso-LiS-core-2.6.17-1.2139_FC5-0.9.2.1.rc2-1.FC5.i686.rpm`'
'`striso-streams-core-2.6.17-1.2139_FC5-0.9.2.1.rc2-1.FC5.i686.rpm`'

'`/lib/modules/2.6.17-1.2139_FC5`'
: This relocatable directory contains the kernel modules that provide the striso STREAMS core, drivers and modules.[28]

'`striso-LiS-info-2.6.17-1.2139_FC5-0.9.2.1.rc2-1.FC5.i686.rpm`'
'`striso-streams-info-2.6.17-1.2139_FC5-0.9.2.1.rc2-1.FC5.i686.rpm`'

'`/usr/include/striso/2.6.17-1.2139_FC5`'
: This relocatable directory contains the kernel module exported symbol information that allows other kernel modules to be compiled against the correct version of the striso package.[29]

'`striso-dev-0.9.2.1.rc2-1.FC5.i686.rpm`'
: (not relocatable)

'`striso-devel-0.9.2.1.rc2-1.FC5.i686.rpm`'

'`/usr/lib`'
: This relocatable directory contains striso libraries.

'`/usr/include/striso`'
: This relocatable directory contains striso header files.

'`striso-doc-0.9.2.1.rc2-1.FC5.i686.rpm`'

'`/usr/share/doc`'
: This relocatable directory contains all package specific documentation (including this manual). The subdirectory in this directory is the '`striso-0.9.2.1.rc2`' directory.

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

---

[28] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

[29] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example. Also, note that the '`info`' subpackage is only applicable to the 2.4 kernel series.

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

‘`striso-LiS-lib-0.9.2.1.rc2-1.FC5.i686.rpm`’
‘`striso-streams-lib-0.9.2.1.rc2-1.FC5.i686.rpm`’

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

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

‘`striso-source-0.9.2.1.rc2-1.FC5.i686.rpm`’

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

‘`striso-LiS-util-0.9.2.1.rc2-1.FC5.i686.rpm`’
‘`striso-streams-util-0.9.2.1.rc2-1.FC5.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 5.4.1 [Installing the Binary RPM], page 46.

5.2.2 Configuring the Debian DEB

In general the binary DEB do not require any configuration.

Installation

To install the Debian DEB, See Section 5.4.2 [Installing the Debian DEB], page 47.

5.2.3 Configuring the Source RPM

When building from the source RPM (see Section 5.3.1 [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 5.3.1 [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 acheive the same effect as:

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

do:

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

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

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

`--with checks`
`--without checks`
: Enable or disable preinstall checks. Each packages supports a number of preinstall checks that can be performed by invoking the '`check`' target with `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.[30] 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 '`striso`' 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

[30] 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.

removing static and inline attributes from functions and invoking all debugging macros in the code. The default is to not perform kernel debugging.

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

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

`--with k-inline`
`--without k-inline`
: Specifies whether kernel `inline` functions are to be 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 5.2.5 [Configuring the Tar Ball], page 32.

`--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 `reuild` target uses this option to signal to the RPM spec file that the '`dev`' subpackage need not be built. This option does not appear when the package has no devices.

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

`--with 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.

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

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

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

Build

To build from the source RPM, See Section 5.3.1 [Building from the Source RPM], page 41.

5.2.4 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 5.2.5 [Configuring the Tar Ball], page 32. For an example, See Section 5.3.2 [Building from the Debian DSC], page 42.

Build

To build from the Debian DSC, See Section 5.3.2 [Building from the Debian DSC], page 42.

5.2.5 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.

5.2.5.1 Configure Options

This is a generic description of common `configure` options. Not all `configure` options are applicable to all release packages. Options that are kernel module specific are only applicable to release packages that build kernel modules. STREAMS options are only applicable to release packages that provide or require STREAMS.

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

`--enable-checks`
`--disable-checks`
: Enable or disable preinstall checks. Each 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 '`striso`' release. No private components exist in '`striso`' releases.

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

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

`--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. This option has no effect if there are no kernel modules in the package.

`--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. This option has no effect if there are no architecture dependent components in the package.

`--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. This options has no effect if there are no architecture independent components in the package.

`--enable-k-inline`
: Enable kernel inline functions. Most Linux kernels build without `-finline-functions`. This option adds the `-finline-functions` and `-Winline` flags to the compilation of kernel modules. Use with care. This option has no effect if there are no kernel modules in the package.

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

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

`--enable-k-debug`
: Enable kernel module run-time debugging. Specifies whether kernel debugging is to be performed. This option is mutuallly 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 debuggin macros in the code (including performance-affecting debug macros). The default is to not perform kernel debugging. This option has no effect if there are no kernel modules in the package.

`--disable-k-modversions`
: Disable module versions on `striso` symbols. Specifies whether kernel symbol versioning is to be used on symbols exported from built `striso` modules. The default is to provide kernel symbol versioning on all exported symbols. This option has no effect if there are no kernel modules in the package.

`--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 `reuild` 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 ther 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.[31] 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 conatined 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`'.

[31] 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-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 int he 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.

`--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 direc-

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

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

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

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

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

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

`--with-k-config=K-CONFIG`
: Specify the '`K-CONFIG`' kernel configuration file. The default is '*`BOOT`*`/config-K-RELEASE`'. This configuration file is normally located by the `configure` script and need only be provided forspecial 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-lis[=LIS-DIR]`
`--without-lis`
: Specify the '`LIS-DIR`' directory in which to find LiS headers. Also specifies that the build is to be made against Linux STREAMS. The default is '`/usr/include/LiS`' if it exists, '`no`' otherwise. This directory is normally located by

the `configure` script and need only be provided for special cross-build environments or when requested by a `configure` script error message. This option has no effect on packages that do not use the STREAMS subsystem.

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

`--with-strconf-master=STRCONF_CONFIG`
: Specify the '`STRCONF_CONFIG`' file name to which the configuration master file is written. The default is '`Config.master`'. This option has no effect on packages that do not use the STREAMS subsystem and the strconf scripts.

`--with-base-major=STRCONF_MAJBASE`
: Start numbering for major devices at '`STRCONF_MAJBASE`'. The default is '`230`'. This option has no effect on packages that do not use the STREAMS subsystem and the strconf scripts.

5.2.5.2 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 pacakges 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 architecutre.

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.

5.2.5.3 Build

To build from the tar ball, See Section 5.3.3 [Building from the Tar Ball], page 42.

5.3 Building

5.3.1 Building from the Source RPM

If you have downloaded the necessary source RPM (see Section 5.1.3 [Downloading the Source RPM], page 23), 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 5.4.1 [Installing the Binary RPM], page 46).

The source RPM is rebuilt to binary RPMs as follows:

```
% wget http://www.openss7.org/rpms/SRPMS/striso-0.9.2.1.rc2-1.src.rpm
% rpmbuild --rebuild -vv striso-0.9.2.1.rc2-1.src.rpm
```

The rebuild process can also recognize a number of options that can be used to tweak the resulting binaries, See Section 5.2.3 [Configuring the Source RPM], page 28. These options are provided on the `rpm` command line. For example:

```
% rpmbuild --rebuild -vv --target athlon-redhat-linux \
  --define "_kversion 2.6.17-1.2139_FC5" \
  --with lis -- striso-0.9.2.1.rc2-1.src.rpm
```

will rebuild binary RPM for the '`2.6.17-1.2139_FC5`' kernel for the '`athlon`' architecture against the *LiS* STREAMS package.[32]

Installation

To install the resulting binary RPM, See Section 5.4.1 [Installing the Binary RPM], page 46.

5.3.2 Building from the Debian DSC

If you have downloaded the necessary Debian DSC (see Section 5.1.4 [Downloading the Debian DSC], page 24), 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 5.4.2 [Installing the Debian DEB], page 47).

The Debian DSC is rebuilt to binary DEBs as follows:

```
% wget http://www.openss7.org/debian/striso_0.9.2.1.rc2-0.dsc
% wget http://www.openss7.org/debian/striso_0.9.2.1.rc2-0.tar.gz
% dpkg-buildpackage -v striso_0.9.2.1.rc2-0.dsc
```

The rebuild process can also recognize a number of options that can be used to tweak the resulting binaries, See Section 5.2.4 [Configuring the Debian DSC], page 32. These options are provided in the environment variable *BUILD_DPKGOPTIONS* and have the same form as the options to `configure`, See Section 5.2.5 [Configuring the Tar Ball], page 32. For example:

```
% BUILD_DEBOPTIONS='
        --with-lis
        --with-k-release=2.6.17-1.2139_FC5
        --host=athlon-debian-linux-gnu'
  dpkg-buildpackage -v \
  striso_0.9.2.1.rc2-0.dsc
```

will rebuild binary DEB for the '`2.6.17-1.2139_FC5`' kernel for the '`athlon`' architecture against the *LiS* STREAMS package.[33]

Installation

To install the resulting binary DEB, See Section 5.4.2 [Installing the Debian DEB], page 47.

5.3.3 Building from the Tar Ball

If you have downloaded the tar ball (see Section 5.1.5 [Downloading the Tar Ball], page 24), then the following instructions will rebuild the package on your system. (Note that the build process does not required `root` privilege.)

[32] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

[33] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

5.3.3.1 Native Build

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

```
% wget http://www.openss7.org/striso-0.9.2.1.rc2.tar.bz2
% tar -xjvf striso-0.9.2.1.rc2.tar.bz2
% pushd striso-0.9.2.1.rc2
% ./configure
% make
% popd
```

5.3.3.2 Cross-Build

Following is an example for a cross-build. The kernel release version must always be specified for a cross-build.[34] 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/striso-0.9.2.1.rc2.tar.bz2
% tar -xjvf striso-0.9.2.1.rc2.tar.bz2
% pushd striso-0.9.2.1.rc2
% ./configure DESTDIR="/some/other/root" \
--with-k-release=2.4.18 --host sparc-linux
% make
% popd
```

5.3.3.3 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`'.)

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

[34] 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.

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 5.1.5 [Downloading the Tar Ball], page 24) and configure (see Section 5.2.5 [Configuring the Tar Ball], page 32) 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.[35]

   Any of the normal `configure` script options (see Section 5.2.5 [Configuring the Tar Ball], page 32) 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 5.4.3 [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`'.)

[35] 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/striso-0.9.2.1.rc2.tar.bz2
% tar -xjvf striso-0.9.2.1.rc2.tar.bz2
% pushd striso-0.9.2.1.rc2
% ./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 '*$NEXUSWARE_PREFIX*`/lib/modules/2.4.18/striso`', libraries installed in '*$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 '`striso`' 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 '`striso`'. 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/striso`', 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 LiS build
#       (cd usr/src/LiS; ./configure.nexusware; $(MAKE) install-strip)
#       uncomment for LfS build
        (cd usr/src/streams; ./configure.nexusware; $(MAKE) install-strip)
#       uncomment for striso build
#       (cd usr/src/striso; ./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/striso-0.9.2.1.rc2.tar.bz2
% pushd /u5/NexusWare24
% source SETUP.sh
% pushd usr/src
% tar -xjvf ${DIRSTACK[2]}/striso-0.9.2.1.rc2.tar.bz2
% ln -sf striso-0.9.2.1.rc2 striso
% popd
% make
% popd
```

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

5.4 Installing

5.4.1 Installing the Binary RPM

If you have downloaded the necessary binary RPMs (see Section 5.1.1 [Downloading the Binary RPM], page 17), or have rebuilt binary RPMs using the source RPM (see Section 5.3.1 [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 striso-*-0.9.2.1.rc2-1.FC5.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' \
        -- striso-doc-0.9.2.1.rc2-1.FC5.i686.rpm
```

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

5.4.2 Installing the Debian DEB

If you have downloaded the necessary Debian DEBs (see Section 5.1.2 [Downloading the Debian DEB], page 20), or have rebuild binary DEBs using the Debian DSC (see Section 5.3.2 [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 striso-*_0.9.2.1.rc2-0_*.deb
```

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

5.4.3 Installing the Tar Ball

After the build process (see Section 5.3.3 [Building from the Tar Ball], page 42), 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.

5.5 Removing

5.5.1 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 '^striso-'`
```

For more information on `rpm`, see section "rpm(8)" in *The Manual Pages*.

5.5.2 Removing the Debian DEB

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

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

For more information on `dpkg`, see section "dpkg(8)" in *The Manual Pages*.

5.5.3 Removing the Source RPM

To remove all the installed binary RPM build from the source RPM, see Section 5.5.1 [Removing the Binary RPM], page 47. Then simply remove the binary RPM package files and source RPM file. A command such as:

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

should remove all '`striso`' RPMs from your system.

5.5.4 Removing the Debian DSC

To remove all the installed binary DEB build from the Debian DSC, see Section 5.5.2 [Removing the Debian DEB], page 47. Then simply remove the binary DEB package files and Debian DSC file. A command such as:

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

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

5.5.5 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/striso
% make uninstall
% cd ..
% rm -fr striso-0.9.2.1.rc2
% rm -f striso-0.9.2.1.rc2.tar.gz
% rm -f striso-0.9.2.1.rc2.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 5.4.3 [Installing the Tar Ball], page 47) except the final installation and then perform the steps above.

5.6 Loading

5.6.1 Normal Module Loading

When '`striso`' 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/2.6.17-1.2139_FC5/striso`'. The '`make install`' process should have copied the kernel module files '`streams-*.o`' to the directory '`/lib/modules/2.6.17-1.2139_FC5/striso`'. This means that to load any of these modules, you can simply execute, for example, '`modprobe stream-`*`somedriver`*'.[36]

5.6.1.1 Linux STREAMS Module Loading

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

The convention for '`striso`' kernel loadable object files is:

- Their name start with "streams-".
- They are placed in '`/lib/modules/2.6.17-1.2139_FC5/streams/`', where '`2.6.17-1.2139_FC5`' is an example kernel version.

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

5.6.1.2 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*.

[36] Note that the '`_kversion`' of '`2.6.17-1.2139_FC5`' is only an example.

5.6.2 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.[37]

5.7 Maintenance

5.7.1 Makefile Targets

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

5.7.1.1 User Targets

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

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

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

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

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

Another way to enable the greater set of checks, without invoking maintainer mode, is to specify '`--enable-checks`' to `configure`.

[37] 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.

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

'`install`'
'`install-strip`'
: The '`install`' target installs the package. This target also performs some actions similar to the pre- and post-install scripts used by packaging tools such as `rpm` or `dpkg`. The '`install-strip`' target strips unnecessary symbols from executables and kernel modules before installing.

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

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

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

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

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

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

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

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

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

‘`remove`’ This target is like ‘`uninstall`’ with the exception that it uninstalls in the reverse order that installation was performed.[40]

This is an *OpenSS7 Project* specific makefile target.

5.7.1.2 Maintainer Targets

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

‘`dist`’ Creates a distribution package (tarball) in the top level build directory. *OpenSS7 Project* packages distribute two archives: a ‘`gzip tar`’ archive and a ‘`bzip tar`’ archive. These archives will have the name ‘`striso-0.9.2.1.rc2.tar.gz`’ and ‘`striso-0.9.2.1.rc2.tar.bz2`’.

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

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

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

5.7.1.3 Clean Targets

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

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

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

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

‘`distclean`’
This target cleans out the directories left behind by ‘`distcheck`’ and removes all the `configure` and generated files from the build directory. This will effectively

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

remove all the files in the build directory, with the except of files that belong to you or some other process.

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

'`maintainer-clean`'
: This target not only removes files from the build directory, it removes generated files from the source directory as well. Care should be taken when invoking this target, because it removes files generated by the maintainer and distributed with the archive that might require special tools to regenerate. These special tools might only be available to the maintainer (but they aren't). It also means that you probably need a full blown Linux system to rebuild the package.

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

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

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

5.7.1.4 Release Targets

The following are targets used to generate complete releases into the package distribution directory. These are good for unattended and NFS builds, which is what I use them for. Also, when building from atop multiple packages, these targets also recurse down through each package.

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

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

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

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

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

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

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

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

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

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

5.7.1.5 Logging Targets

For convenience, to log the output of a number of targets to a file, log targets are defined. The log file itself is used as the target to make, but make invokes the target minus a '`.log`' suffix. So, for example, to log the results of target '`foo`', invoke the target '`foo.log`'. The only target that this does not apply to is '`compile.log`'. When you invoke the target '`compile.log`' a simple `make` is invoked and logged to the file '`compile.log`'. The '`foo.log`' rule applies to all other targets. This does not work for all targets, just a selected few.[41] Following are the logging targets:

Common Logging Targets

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

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

'`check.log`'
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake` makefile target '`check`'.

'`install.log`'
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake` makefile target '`install`'.

'`installcheck.log`'
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake` makefile target '`installcheck`'.

'`uninstall.log`'
: This is an *OpenSS7 Project* specific makefile target, but it invokes the standard *GNU* `automake` makefile target '`uninstall`'.

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

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

Maintainer Logging Targets

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

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

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

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

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

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

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

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

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

5.7.1.6 Problem Report Targets

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

There are two additional problem report targets:

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

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

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

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

5.7.1.7 Release Archive Targets

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

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

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

The files generated are named:

'`striso-0.9.2.1.rc2.tar.gz`' and '`striso-0.9.2.1.rc2.tar.bz2`'

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

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

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

'`make GNUPGPASS=mypasswd release-sign-archives`'

Signature files will be named:

'`striso-0.9.2.1.rc2.tar.gz.asc`' and '`striso-0.9.2.1.rc2.tar.bz2.asc`'

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

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

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

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

5.7.1.8 RPM Build Targets

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

'`srpm`' This target generates the source rpm for the package (without signing the source rpm). The source rpm will be named: '`striso-0.9.2.1.rc2-1.srpm`'.

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

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

'`striso-*-0.9.2.1.rc2-1.*.rpm`'

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

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

'`sign`'
'`srpm-sign`'
These two targets are the same. When invoked, they will add a signature to the source rpm file, provided that the file does not already have a signature. You will be prompted for a password if a signature is required. Automated or unattended builds can be acheived by using the `emake` expect script, included in '`${srcdir}/scripts/emake`'.

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

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

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

'`resign`' This target will search out and sign, with a *GPG* signature, the source rpm, and all of the binary rpms for this package that can be found in the package distribution directory. This target will prompt for a *GPG* password. Automated or

unattended builds can be acheived with the `emake` expect script loccated here: '`${srcdir}/scripts/emake`'.

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

5.7.1.9 Debian Build Targets

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

'`dsc`' This target will build the debian source change package ('`.dsc`' file). We use release number '`0`' so that the entire tarball is included in the '`dsc`' file. You can use release number '`1`' for the same purposes. Release numbers above '`1`' will not include the entire tarball. The '`.dsc`' file will be named: '`striso_0.9.2.1.rc2-0.dsc`'.

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

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

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

'`debs`' This target will build the debian binary package ('`.deb`' file) from the '`.dsc`' created above. (This target will also create the '`.dsc`' if it has not been created already.) The subpackage '`.deb`' files will be named: '`striso-*_0.9.2.1.rc2-0_*.deb`', where the stars indicate the subpackage and the architecture.

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

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

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

6 Troubleshooting

6.1 Test Suites

6.1.1 Pre-installation Checks

Most *OpenSS7* packages, including the *OpenSS7 ISO Stack* 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 5.3.1 [Building from the Source RPM], page 41; and Section 5.3.2 [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 automagically 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 5.3.3 [Building from the Tar Ball], page 42; and Section 5.4.3 [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 6.1.

```
% wget http://www.openss7.org/striso-0.9.2.1.rc2.tar.bz2
% tar -xjvf striso-0.9.2.1.rc2.tar.bz2
% pushd striso-0.9.2.1.rc2
% ./configure
% make
% make check  # <------- invoke pre-installation checks
% popd
```

Example 6.1: *Invoking Pre-Installation Checks*

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

6.1.1.1 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 6.2 will perform *System* checks.

```
% wget http://www.openss7.org/striso-0.9.2.1.rc2.tar.bz2
% tar -xjvf striso-0.9.2.1.rc2.tar.bz2
% pushd striso-0.9.2.1.rc2
% ./configure
% make
% make check  # <------ invokes System pre-installation checks
% popd
```

Example 6.2: *Invoking System Checks*

6.1.1.2 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 6.3 will perform *Maintenance* checks.

```
% wget http://www.openss7.org/striso-0.9.2.1.rc2.tar.bz2
% tar -xjvf striso-0.9.2.1.rc2.tar.bz2
% pushd striso-0.9.2.1.rc2
% ./configure --enable-maintainer-mode
% make
% make check  # <------ invokes Maintenance pre-installation checks
% popd
```

Example 6.3: *Invoking Maintenace Checks*

6.1.1.3 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 *Maintainance* checks.

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

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

6.1.2 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.

6.1.2.1 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 6.4.

```
% wget http://www.openss7.org/striso-0.9.2.1.rc2.tar.bz2
% tar -xjvf striso-0.9.2.1.rc2.tar.bz2
% pushd striso-0.9.2.1.rc2
% ./configure
% make
% make check  # <------ invokes System pre-installation checks
% make install
% sudo make installcheck # <------- invokes post-installation tests
% popd
```

Example 6.4: *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 depositied upon test suite failure in the '`tests`' directory. This '`testsuite.log`' file can then be attached as part of the problem report and provides rich details to maintainers of the package. See also See Section 6.2 [Problem Reports], page 63, below.

Typical steps for invoking and installed `testsuite` standalone are shown in Example 6.5.

```
% [sudo] /usr/libexec/striso/testsuite
```

Example 6.5: *Invoking* `testsuite` *Directly*

When invoked directly, `testsuite` will generate a '`testsuite.log`' file in the current directory, and a '`testsuite.dir`' directory of failed tests cases and debugging scripts. For generating a problem report for failed test cases, see Section 6.2.4 [Stand Alone Problem Reports], page 66.

6.2 Problem Reports

6.2.1 Problem Report Guidelines

Problem reports in the following categories should include a log file as indicated in the table below:

‘`./configure`’
: A problem with the configuration process occurs that causes the ‘`./configure`’ command to fail. The problem report must include the ‘`config.log`’ file that was generated by `configure`.

‘`make compile.log`’
: A problem with the build process occurs that causes the ‘`make`’ command to fail. Perform ‘`make clean`’ and then ‘`make compile.log`’ and attach the ‘`config.log`’ and ‘`compile.log`’ files to the problem report.

‘`make check.log`’
: A problem occurs with the ‘`make check`’ target that causes it to fail. Perform ‘`make check-clean check.log`’ and attach the ‘`config.log`’, ‘`compile.log`’ and ‘`check.log`’ files to the problem report.

‘`sudo make install.log`’
: A problem occurs with ‘`sudo make install`’ that causes it to fail. Perform ‘`sudo make uninstall`’ and ‘`sudo make install.log`’ and attach the ‘`config.log`’, ‘`compile.log`’, ‘`check.log`’, and ‘`install.log`’ files to the problem report.

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

‘`[sudo] make uninstall.log`’
: A problem occurs with the ‘`make uninstall`’ target that causes the test suite to fail. Perform ‘`sudo make uninstall.log`’ and attach the ‘`config.log`’, ‘`compile.log`’, ‘`check.log`’, ‘`install.log`’, ‘`installcheck.log`’, ‘`tests/testsuite.log`’ and ‘`uninstall.log`’ file to the problem report.

‘`[sudo] make remove.log`’
: A problem occurs with the ‘`make remove`’ target that causes the test suite to fail. Perform ‘`sudo make remove.log`’ and attach the ‘`config.log`’, ‘`compile.log`’, ‘`check.log`’, ‘`install.log`’, ‘`installcheck.log`’, ‘`tests/testsuite.log`’ and ‘`remove.log`’ file to the problem report.

For other problems that occur during the use of the *OpenSS7 ISO Stack* 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)`).

6.2.2 Generating Problem Reports

The OpenSS7 Project uses the *GNU GNATS* system for problem reporting. Although the ‘`send-pr`’ tool from the *GNU GNATS* package can be used for bug reporting to the project’s

GNATS database using electronic mail, it is not always convenient to download and install the *GNATS* system to gain access to the '`send-pr`' tool.

Therefore, the *OpenSS7 ISO Stack* package provides the '`send-pr`' shell script that can be used for problem reporting. The '`send-pr`' shell script can invoked directly and is a workalike for the *GNU* '`send-pr`' tool.

The '`send-pr`' tool takes the same flags and can be used in the same fashion, however, whereas '`send-pr`' is an interactive tool[2], '`send-pr`' is also able to perform batch processing. Whereas '`send-pr`' takes its field information from local databases or from using the '`query-pr`' C-language program to query a remote database, the '`send-pr`' tool has the field database internal to the tool.

Problem reports can be generate using `make`, See Section 5.7.1.6 [Problem Report Targets], page 55. An example of how simple it is to generate a problem report is illustrated in Example 6.6.

```
% make pr
SEND-PR:
SEND-PR: send-pr:  send-pr was invoked to generate an external report.  An
SEND-PR: automated problem report has been created in the file named
SEND-PR: 'problem.pr' in the current directory.  This problem report can
SEND-PR: be sent to bugs@openss7.org by calling this script as
SEND-PR: '/home/brian/os7/scripts/send-pr --file="problem.pr"'.
SEND-PR:
SEND-PR: It is possible to edit some of the fields before sending on the
SEND-PR: problem report.  Please remember that there is NO WARRANTY.  See
SEND-PR: the file 'COPYING' in the top level directory.
SEND-PR:
SEND-PR: Please do not send confidential information to the bug report
SEND-PR: address.  Inspect the file 'problem.pr' for confidential
SEND-PR: information before mailing.
SEND-PR:
% vim problem.pr  # <--- follow instructions at head of file
% make send-pr
```

Example 6.6: *Invoking Problem Report Generation*

Using the '`make pr`' target to generate a problem report has the advantages that it will assemble any available '`*.log`' files in the build directory and attach them to the problem report.

6.2.3 Automatic Problem Reports

The *OpenSS7 ISO Stack* package also provides a feature for automatic problem report generation that meets the problem report submission guidelines detailed in the preceding sections.

Whenever a logging makefile target (see Section 5.7.1.5 [Logging Targets], page 54) is invoked, if the primary target fails, the `send-pr` shell script is invoked to automatically

[2] '`send-pr`' launches the user's *EDITOR* to edit the problem report before submitting it.

generate a problem report file suitable for the corresponding target (as described above under see Section 6.2.1 [Problem Report Guidelines], page 64). An example is shown in Example 6.8.

```
% make compile.log
...
...
make[5]: *** [libXNSdrvs_a-ip.o] Error 1
make[5]: Leaving directory `/u6/buildel4/strxns'
make[4]: *** [all-recursive] Error 1
make[4]: Leaving directory `/u6/buildel4/strxns'
make[3]: *** [all] Error 2
make[3]: Leaving directory `/u6/buildel4/strxns'
make[2]: *** [all-recursive] Error 1
make[2]: Leaving directory `/u6/buildel4'
make[1]: *** [all] Error 2
make[1]: Leaving directory `/u6/buildel4'
SEND-PR:
SEND-PR: send-pr:  Make target compile.log failed in the compile stage.  An
SEND-PR: automated problem report has been created in the file named
SEND-PR: 'problem.pr' in the current directory.  This problem report can
SEND-PR: be sent to bugs@openss7.org by calling 'make send-pr'.
SEND-PR:
SEND-PR: It is possible to edit some of the fields before sending on the
SEND-PR: problem report.  Please remember that there is NO WARRANTY.  See
SEND-PR: the file 'COPYING' in the top level directory.
SEND-PR:
SEND-PR: Please do not send confidential information to the bug report
SEND-PR: address.  Inspect the file 'problem.pr' for confidential
SEND-PR: information before mailing.
SEND-PR:
% vim problem.pr  # <--- follow instructions at head of file
% make send-pr
```

Example 6.7: *Problem Report from Failed Logging Target*

6.2.4 Stand Alone Problem Reports

The *OpenSS7 ISO Stack* package installs the `send-pr` script and its configuration file '`send-pr.config`' in '`${libexecdir}/striso`' along with the validation `testsuite`, see See Section 6.1 [Test Suites], page 59. As with the `testsuite`, this allows the `send-pr` script to be used for problem report generation on an installed system that does not have a build directory.

An example of invoking the package `testsuite` and then generating a problem report for failed cases is shown in Example 6.8.

```
% [sudo] /usr/libexec/striso/testsuite
% # test cases failed...
% /usr/libexec/striso/send-pr
SEND-PR:
SEND-PR: send-pr:  send-pr was invoked to generate an external report.  An
SEND-PR: automated problem report has been created in the file named
SEND-PR: 'problem.pr' in the current directory.  This problem report can
SEND-PR: be sent to bugs@openss7.org by calling this script as
SEND-PR: '/usr/libexec/striso/send-pr --file problem.pr'.
SEND-PR:
SEND-PR: It is possible to edit some of the fields before sending on the
SEND-PR: problem report.  Please remember that there is NO WARRANTY.  See
SEND-PR: the file 'COPYING' in the top level directory.
SEND-PR:
SEND-PR: Please do not send confidential information to the bug report
SEND-PR: address.  Inspect the file 'problem.pr' for confidential
SEND-PR: information before mailing.
SEND-PR:
% vim problem.pr  # <--- follow instructions at head of file
% /usr/libexec/striso/send-pr --file problem.pr
```

Example 6.8: *Invoking* `send-pr` *Directly*

The advantage of the approach shown in the example is that the `send-pr` script is capable of collecting the '`testsuite.log`' file and the failed test cases and debugging scripts from the '`testsuite.dir`' directory and including them in the problem report, as well as all package pertinent information from the installed '`send-pr.config`'.

6.3 Known Problems

The OpenSS7 Project does not ship software with known bugs. All bugs are unknown.

Verified behaviour is that behaviour that has been verified by conformance test suites that are shipped with the *OpenSS7 ISO Stack* package.

Unverified behaviour may contain unknown bugs.

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

See also Section 4.5 [Bugs], page 15.

Appendix A GPL

A.1 GNU General Public License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

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 LGPL

B.1 GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

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

B.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 Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

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

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

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

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

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

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

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

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

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

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

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

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

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

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

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

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be

distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

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

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

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

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

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

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

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

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

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

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

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

      (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d

requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

   c. Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

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

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

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

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

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:
   a. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
   b. Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.
10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.
11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent

obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

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

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

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

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

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

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

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

NO WARRANTY

15. **BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MER-**

CHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

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

END OF TERMS AND CONDITIONS

B.1.3 How to Apply These Terms to Your New Libraries

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

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

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

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

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

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

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

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

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

signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!

Appendix C FDL

C.1 GNU Free Documentation License

GNU FREE DOCUMENTATION LICENSE
Version 1.1, March 2000

C.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. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

C.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

C.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.

Appendix D Indices

D.1 Index of Concepts

N

O

P

R

S

T

U

D.2 Index of Data Types

(Index is nonexistent)

D.3 Index of Functions and Macros

(Index is nonexistent)

D.4 Index of Variables and Constants

D.5 Index of Files and Programs

D.6 Index of Configuration Options

Index of Makefile Targets

D.8 Index of Authors

(Index is nonexistent)