ZebOS-XP®
Network Platform

Version 1.4
Extended Performance

ZebOS-XP Install Guide

December 2015

IP Infusion Inc.
3965 Freedom Circle, Suite 200
Santa Clara, CA 95054
+1 408-400-1900
http://www.ipinfusion.com/

For support, questions, or comments via E-mail, contact:
support@ipinfusion.com

Contents

Preface

This guide show you how to configure, build, and run ZebOS-XP.

Audience

This guide is intended for developers who build ZebOS-XP.

At a minimum, you need to be familiar with:

- Linux operating systems and their command line interfaces
- How to configure a working build environment for the intended target, for example, a server, a router or a switch
- The cross-compilation process using `configure`
- Internet routing and switching software protocols

Conventions

Table P-1 shows the conventions used in this guide.

Table P-1: Conventions

Convention	Description
Italics	Emphasized terms; titles of books
Note:	Special instructions, suggestions, or warnings
`monospaced type`	Code elements such as commands, functions, parameters, files, and directories
`[TERM]`	Brackets enclose terms for which you substitute an appropriate value. For example, in the command below, you substitute a directory path, and not the word `PATH`: `patch -p0 [PATH]/ZebOS/...`

Contents

This document contains these chapters and appendices:

- Appendix E, *Building ZebOS-XP for Wind River Linux*
- Appendix F, *Copyright Information*

Related Documents

The following guides are related to this document:

- *Architecture Guide*

Note: All ZebOS-XP technical manuals are available to licensed customers at http://www.ipinfusion.com/support/document_list.

Support

For support-related questions, contact support@ipinfusion.com.

Comments

If you have comments, or need to report a problem with the content, contact techpubs@ipinfusion.com.

CHAPTER 1 Building ZebOS-XP for Fedora 17

This chapter contains all of the information needed to prepare for and compile ZebOS-XP with a Linux 3.4 kernel on Fedora 17. For a first-time installation, perform all steps. As your system expands, compile the software, and make the target. For more information on customizing protocols, refer to the ZebOS-XP developer guides.

Note: This installation is not shell-dependent; use any convenient Linux shell.

Building Linux

This section describes how you need to configure and build Linux for ZebOS-XP. Before building ZebOS-XP, make sure that you can build Linux on a computer with an x86 processor.

Download Linux

ZebOS-XP supports the following version of the Linux kernel:

- Linux 3.4.44

This version of ZebOS-XP has been verified to work with the Linux distribution cited in the *Release Notes* for version 3.4.44 of the Linux kernel.

To get this version of the Linux kernel:

1. Download `linux-3.4.44.tar.bz2` from https://www.kernel.org/pub/linux/kernel/v3.x/.
2. Untar the Linux 3.4.44 kernel source code:

```
cd /usr/src/
tar xvfj linux-3.4.44.tar.bz2
```

Configure the Linux Kernel

This section contains procedures for configuring the Linux kernel to support ZebOS-XP.

1. Change to the Linux kernel source code folder:

```
cd /usr/src/linux-3.4.44
```

2. Open the kernel configuration window:

```
make mrproper   # Delete any unwanted configuration
make oldconfig  # Use current host configuration (optional)
make menuconfig # Configure the kernel
```

Note: Give the `make help` command to get a list of all possible options.

3. Review the Linux kernel configuration options in Table 1-1. These options need to be enabled to build ZebOS-XP. In this version of the Linux kernel they are enabled by default and you do not need to take any action.

Table 1-1: Linux kernel configuration options

Option	Description
CONFIG_EXPERIMENTAL	Prompt for development and/or incomplete code/drivers
CONFIG_PACKET	Packet socket
CONFIG_UNIX	Unix domain sockets
CONFIG_NET_IPIP	IP: tunneling
COFNIG_NET_IPGRE	IP: GRE tunnels over IP
CONFIG_NET_IPGRE_BROADCAST	IP: broadcast GRE over IP
CONFIG_VLAN_8021Q	802.1Q VLAN Support
CONFIG_IPV6	The IPv6 protocol
CONFIG_IPV6_SIT	IPv6: IPv6-in-IPv4 tunnel (SIT driver)
CONFIG_IP_MULTICAST	IP: multicasting
CONFIG_IP_MROUTE	IP: multicast routing
CONFIG_IP_PIMSM_V1	IP: PIM-SM version 1 support
CONFIG_IP_PIMSM_V2	IP: PIM-SM version 2 support
CONFIG_IPV6_MROUTE	IPv6: multicast routing
CONFIG_IPV6_PIMSM_V2	IPv6: PIM-SM version 2 support
CONFIG_TCP_MD5SIG	TCP: MD5 Signature Option support (RFC 2385)
CONFIG_TIPC = m	
CONFIG_TIPC_ADVANCED = y	
CONFIG_TIPC_PORTS = 8191	Transparent Inter-process Communication (TIPC) support on kernel
CONFIG_TIPC_LOG = 0	
CONFIG_TIPC_DEBUG = y	

4. Review the Linux kernel configuration option in Table 1-3. This option will provide additional kernel information.If it needs to be disabled, we have the option to do so. In this version of the Linux kernel, this option is enabled by default

Table 1-2: Linux kernel configuration option for dynamic debugging

Option	Description
CONFIG_DYNAMIC_DEBUG	Dynamic `printk` support

5. Review the Linux kernel configuration option to enable SMP in Table 1-3. This option needs to be enabled to build SMP in ZebOS-XP. In this version of the Linux kernel, this option is enabled by default.

Table 1-3: Linux kernel configuration option for SMP

Option	Description
CONFIG_SMP	Symmetric multi-processing support

   Explicitly disable the kernel option in Table 1-3 to disable support for SMP:

6. Select the appropriate drivers for your network interface card.
7. Save the kernel configuration and exit.
8. Proceed to Compile the Linux Kernel on page 11.

Compile the Linux Kernel

This section shows how to build Linux from the source that you downloaded and configured.

Note: Before compiling the kernel, you must also apply patches as explained in Apply Kernel Patches.

1. Download the Linux development tools using your distribution's package management tool.
2. Change to the Linux kernel source code folder:
   ```
   cd /usr/src/linux-3.4.44
   ```
3. Build the kernel using the make utility:
   ```
   make
   ```
4. Install the kernel by entering these commands:
   ```
   make modules_install
   make install
   ```

Building ZebOS-XP

Building ZebOS-XP on a Linux system consists of extracting the source files from the tar file, compiling the source, and installing the object files. Before compilation, you must apply ZebOS-XP-specific patches for the Linux system.

After the build process is complete, the build engine copies the executables to `/usr/local/sbin` and the sample configuration files to /`usr/local/etc`.

Extract ZebOS-XP Modules

1. Change to the directory where you want to store the ZebOS-XP source:
   ```
   cd /myZebOSpath
   ```
2. Extract the ZebOS-XP source software from the `.tar` file:
   ```
   tar -xvzf TARFILE
   ```
   where `TARFILE` is the name of the ZebOS-XP `.tar` file.

Apply Kernel Patches

Table 1-4 lists the patches for Linux version 3.4.44 that are provided by IP Infusion Inc. in `ZebOS/kernel/linux/3.4.44:`

Table 1-4: ZebOS-XP Linux kernel patches

Protocol	Patch Name	Description
Layer-2	`0001-Changes-for-Layer2-software-forwarder.patch`	Layer 2 Forwarding support for the Linux kernel
MPLS	`0002-Changes-for-MPLS-software-forwarder.patch`	MPLS Forwarder support for the Linux kernel
VRRP	`0003-Changes-for-VRRP-support.patch`	VRRP support for the Linux kernel
PIM	`0004-Changes-for-MCAT-bdir.patch`	PIM bidirectional support for the Linux kernel
Multicast	`0005-Changes-for-Multicast-Data-Forwarding-Based-on-Mroute.patch`	Multicast forwarding (both v4 and v6) based on *,G Mroute entries for the Linux kernel
IGMP	`0006-Changes-for-IGMP-packet-processing.patch`	IGMP packet processing to the Linux kernel
MPLS	`0007-Changes-for-OAM-Address-Family.patch`	MPLS OAM address family for the Linux kernel
MLAG	`0008-Changes-for-supporting-MLAG.patch`	Multi-Chassis Link Aggregation Group support for the Linux kernel
BGP-SSO	`0009-Changes-for-supporting-TCP-repair.patch`	Kernel support for migration of the TCP socket for BGP SSO
BGP & LDP	`0010-Changes-for-TCP-MD5-Authtentication.patch`	Unwanted type-casting is removed for using TCP-MD5 authentication
MPLS	`0011-Changes-for-Adding-Werror-in-MPLS.patch`	Avoid Compiler warnings in the MPLS modules
MPLS	`0012-Changes-for-MPLS-rearchitected-software-forwarder.patch`	Enhanced version of the MPLS forwarder

1. Change to the source directory:

   ```
   cd /usr/src
   ```

2. Backup your kernel source:

   ```
   tar cvfz linux-3.4.44.tar.bz2 linux-3.4.44
   ```

3. Change to your kernel source directory:

   ```
   cd /usr/src/linux-3.4.44
   ```

   This assumes that you have set this directory as a soft link pointing to `linux-3.4.44`.

4. Copy the patch to the kernel source directory:

   ```
   cp /PATH/PATCH ./
   ```

   where

   `PATH` is the location of the patch (`ZebOS/kernel/linux/3.4.44`).

   `PATCH` is the name of the patch as shown in Table .

For example, to copy the MPLS patch:

```
cp /myZebOSpath/ZebOS/kernel/linux/3.4.44/0002-Changes-for-MPLS-software-forwarder.patch
./
```

5. Apply the kernel patches in Table 1-4 you need to your Linux source code. For example, to apply the patches for MPLS:

   ```
   patch -Np1 -i 0002-Changes-for-MPLS-software-forwarder.patch
   ```

6. Review the ZebOS-XP-specific Linux kernel configuration options in Table 1-5 for the patch. Refer to Configure the Linux Kernel on page 9. These are enabled as loadable modules by default and you do not need to take any action.

Table 1-5: Linux kernel configuration options enabled as modules

Option	Description
CONFIG_IPIFWD	VLAN-Aware Bridge Forwarder
CONFIG_8021X	802.1x Port Authentication Entity
CONFIG_LACP	Link Aggregation Control Protocol
CONFIG_LLDP	Link Layer Discovery Protocol

7. Recompile the kernel. Refer to Compile the Linux Kernel on page 11 for instructions.

8. Reboot the machine with the new kernel.

Configuration Options

Appendix A, *Build Configuration Options* lists all of the options that can be enabled or disabled. As you proceed through the next two sections, you may wish refer to this appendix for descriptions of each option.

Note: Configuration options can only be enabled if the related ZebOS-XP module were licensed and purchased.

Configuration Shell Script

After the kernel source is patched and rebuilt, the next step is to configure the ZebOS-XP build environment.

IP Infusion Inc. provides a shell script named `config.sh` with ZebOS-XP. The `config.sh` script enables and disables configuration options depending on the ZebOS-XP modules that you purchased. IP Infusion Inc. recommends that you verify the configuration options *before* running the script.

The `config.sh` script tries to determine correct values for various system-dependent variables used during compilation. It might create one or more `.h` files containing system-dependent definitions.

Table 1-6 describes output files that `config.sh` creates.

Table 1-6: ZebOS-XP build output files

File Name	Description
`compile.out`	Compiler output that is useful for debugging.
`config.log`	Messages generated by the compiler.
`builds.log`	The Bill of Materials (BOM) for your ZebOS-XP software, including information about the operating system, protocols, directory structure, and files.

Configure the Build Environment

There are two ways to configure the build environment:

- Automatically by running the `config.sh` script provided by IP Infusion Inc.
- Manually by setting each build option individually

Configure the Build Environment Automatically

The ZebOS-XP configuration script automatically detects most host configurations. Perform the following steps to configure the build environment. Running the `configure` process takes some time. While running, the script generates messages explaining which features it is checking.

1. Verify that all enabled and disabled configuration options match your requirements.
2. Change to the directory that contains the package's source code:
   ```
   cd /myZebOSpath/ZebOS
   ```
3. Create a symbolic link to the Linux kernel source which is used by the `setup.sh` script:
   ```
   ln -sf /usr/src/linux-3.4.44 linux-source
   ```
4. Source the `setup.sh` script provided by IP Infusion Inc. which sets shell variables:
   ```
   source setup.sh
   ```
5. Run the `config.sh` script provided by IP Infusion Inc.:
   ```
   config.sh
   ```
   This script executes the `configure` command and enables or disables configuration options depending on the ZebOS-XP modules purchased.

Configure Build Environment Manually

To set configuration options manually, review the `config.sh` script that came with your ZebOS-XP software before proceeding. Remember, some configuration options are already enabled, depending on the software you have purchased.

1. Change to the directory containing the ZebOS-XP source code:
   ```
   cd /myZebOSpath/ZebOS
   ```
2. Create a symbolic link to the Linux kernel source which is used by the `setup.sh` script:
   ```
   ln -sf /usr/src/linux-3.4.44 linux-source
   ```
3. Source the `setup.sh` script provided by IP Infusion Inc. which sets shell variables:
   ```
   source setup.sh
   ```
4. Choose the configure options to disable. Refer to the `config.sh` script to see the options enabled by default.
5. Choose the configuration options to enable from the tables in Appendix A, *Build Configuration Options*.

   Some options have dependencies, so if all the required options are not enabled for an affected module, you might get an error when you run the `make` command.

   For example, if you have enabled the OSPFv6 option, but did not enable the IPv6 option, then you run the `make` command, you might see:

   ```
   make ospf6d nsm
   Make aborted: Unable to find ../../ospf6d/Makefile
   make: *** [ospf6d] Error 1
   ```

6. Configure the package for the system:

   ```
   ./configure <options>
   ```

   where `<options>` are any build configuration options that you have chosen to enable or disable, for example:

   ```
   --enable-nsm --disable-ipv6
   ```

   The following example shows the output:

```
./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for style of include used by make... GNU
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
...

ZebOS configuration
-------------------
ZebOS version           : 1.1
Build date              : 10/15/13 10:49:27
host operating system   : linux-gnu
source code location    : .
compiler                : gcc
compiler flags          : -m32 -g -O0
directory for pid files : /var/run
```

Often-Used Options

This section describes often-used configuration options.

Layer-2 Software Forwarder

Use the following configuration options to enable the Software Forwarder provided by IP Infusion Inc.:

```
--enable-hal
--enable-swfwdr
```

MPLS Forwarder

Use the following configuration option to enable the MPLS Forwarder provided by IP Infusion Inc.:

```
--enable-mpls-fwd
```

Virtual Private Network (VPN)

To build the VPN feature and enable VRF, enable the following options while compiling:

`--enable-vrf`

`--enable-mpls-fwd` Applicable only when using the ZebOS-XP MPLS Forwarder

Note: To establish a Virtual Private Network, the BGP, OSPF (or RIP), LDP, and NSM daemons must be running.

Virtual Routing (VR)

For Virtual Routing, enable the following configuration option while compiling:

```
--enable-vr
```

Note: Enabling VR automatically enables VRF (Virtual Routing Forwarding) support for the Network Services Manager (NSM), and VR support for LIB, NSM, OSPF and BGP.

Set Up the System for Virtual Routing

Ensure that your system is installed with all of the Advanced Routing options turned on. For testing purposes, check to see that the IPROUTE2 utility package is installed.

High Availability (HA)

To enable HA, refer to the information below.

- For High Availability features, enable the following option:
  ```
  --enable-opensaf
  ```
- To use the Fault Management Simulation, enable the following option:
  ```
  --enable-fm-sim
  ```
- To run HA in debugging mode, enable the following option:
  ```
  --enable-zebos-debug
  ```

Compilation Requirements

To build ZebOS-XP, you need the following versions of Linux development tools:

- `gnu make` version 3.81
- `gcc` version 4.4.6
- `autoconf` version 2.63
- binutils version 2.20.x
- glibc version 2.12

Compile ZebOS-XP

After configuring the build environment, compile ZebOS-XP.

1. Change to the directory containing the ZebOS-XP source code:
   ```
   cd platform/linux
   ```
2. Run the `make clean` utility:
   ```
   make clean
   ```
3. Run the `make` utility. Run `make all` to compile all available modules, or just compile specific protocols by specifying the protocol name. Note that `make install` implies `make all`.

   For example:

   `make <protocol>|all` Makes all modules for the selected `<protocol>`

`make bgpd` Makes the BGP modules
`make install` Makes all modules and daemons

4. If you choose to make individual modules, and do not run the `make all` or `make install` command, and you need the MPLS Forwarder, run the `make all-mpls` command to build it.

5. Likewise if you choose to make individual modules and you need Layer 2 support, run `make all-layer2` to make the Layer 2 support modules.

   Note: The `make` command displays available make targets. You can make individual targets in separate directories by entering `make target-directory`.

   For example:

   `make all-pal` makes all PAL-related files in the PAL target directory
   `make all-lib` makes all library files in the LIB directory
   `make nsm` makes the NSM daemon

Table 1-7 lists the `makefile` commands and their application.

Table 1-7: Makefile commands

Make Targets	Function	Comments
`distclean`	Removes all working files not in the distribution	Applies only at the top level
`install`	Makes and Installs binaries and samples	Applies only at the top level
`all`	Makes all available modules	Applies to the top level and to all available modules
`dep`	Makes all dependencies	Applies to the top level and to all available modules
`forcedep`	Forces all dependencies to be remade	Applies to the top level and to all available modules
`clean`	Cleans up from a previous make	Applies to the top level and to all available modules
<protocol>	Makes the protocol daemon(s) for ZebOS-XP modules	`authd, bgpd, elmid, gvrpd, isisd, isis6d, lagd, ldpd, lmpd, mstpd, nsm, oamd, onmd, ospfd, ospf6d, pimd, pim6d, ripd, ripngd, rstpd, rsvpd, stpd`

Configure Host and Target for Cross-Compiling

Use the `configure` command with the following options when cross-compiling. For example:

```
./configure --host=<xxx> --build=<xxx> --target=<xxx>
```

where <xxx> represents the names of the host, build and target machines, for example:

```
./configure \
--host=i686-redhat-linux-gnu \
--target=powerpc-wrs-linux-gnu \
--build=powerpc-wrs-linux-gnu
```

Build Loadable Kernel Modules

Now that ZebOS-XP is installed and configured, specific kernel modules must be built. Follow the steps in the two sections that follow.

Build Layer 2 Kernel Modules

1. Enable the following configuration options to enable the Software Forwarder provided by IP Infusion Inc.:

   ```
   --enable-hal
   --enable-swfwdr
   ```

2. Change to the `platform` directory:

   ```
   cd platform/linux
   ```

3. Run the following `make` utility to build the Layer 2 module:

   ```
   make all-layer2
   ```

   This utility builds a `layer2_module.ko` file in the `platform/linux/bin` directory.

4. Install the Layer 2 module using the appropriate command for your environment in the `platform/linux/bin` directory:

   ```
   insmod layer2_module.ko
   ```

5. Execute the appropriate command for your environment to ensure that the Layer 2 kernel module is gracefully removed in the event of a reboot or shutdown:

   ```
   rmmod layer2_module.ko
   ```

Build MPLS Kernel Modules

1. Enable the following configuration option to enable the MPLS Forwarder provided by IP Infusion Inc.:

   ```
   --enable-mpls-fwd
   ```

2. Change to the `platform` directory:

   ```
   cd platform/linux
   ```

3. Run the `make` utility to build the MPLS module:

   `make all` (or)

   `make all-mpls` Use this command if you do not want to run `make all`.

   This builds the or the `mpls_module.ko` file in the `platform/linux/bin` directory.

4. Install the MPLS module using the appropriate command for your environment in the `platform/linux/bin` directory:

   ```
   insmod mpls_module.ko
   ```

5. Execute the appropriate command for your environment to ensure that the MPLS Forwarder kernel module is gracefully removed in the event of a reboot or shutdown:

   ```
   rmmod mpls_module.ko
   ```

CHAPTER 2 Building ZebOS-XP for Debian 8

This appendix contains the information needed to prepare and compile ZebOS-XP with a Linux 3.16 kernel on Debian 8. For a first-time installation, perform all steps. As your system expands, compile the software and make the target. For more information on customizing protocols, refer to the ZebOS-XP developer guides.

Note: This installation is not shell-dependent; use any convenient Linux shell.

Building Linux

Please refer to Appendix C, *Install and Build Debian 8*.

Building ZebOS-XP

Building ZebOS-XP on a Linux system consists of extracting the source files from the tar file, compiling the source, and installing the object files. Before compilation, you must apply ZebOS-XP-specific patches for the Linux system.

After the build process is complete, the build engine copies the executables to `/usr/local/sbin` and the sample configuration files to /usr/local/etc.

Extract ZebOS-XP Modules

1. Change to the directory where you want to store the ZebOS-XP source:

   ```
   cd /myZebOSpath
   ```

2. Extract the ZebOS-XP source software from the.tar file:

   ```
   tar -xvzf TARFILE
   ```

   where TARFILE is the name of the ZebOS-XP .tar file.

Configuration Options

Appendix A, *Build Configuration Options* lists the options you can enable or disable. As you proceed through the next two sections, you may wish refer to this appendix for descriptions of each option.

Note: Configuration options can only be enabled if the related ZebOS-XP module were licensed and purchased.

Configuration Shell Script

After the kernel source is patched and rebuilt, the next step is to configure the ZebOS-XP build environment.

IP Infusion Inc. provides a shell script named `config.sh` with ZebOS-XP. The `config.sh` script enables and disables configuration options depending on the ZebOS-XP modules that you purchased. IP Infusion Inc. recommends that you verify the configuration options *before* running the script.

The `config.sh` script tries to determine correct values for various system-dependent variables used during compilation. It might create one or more `.h` files containing system-dependent definitions.

Table 2-1 describes output files that `config.sh` creates.

Table 2-1: ZebOS-XP build output files

File Name	Description
`compile.out`	Compiler output that is useful for debugging.
`config.log`	Messages generated by the compiler.
`builds.log`	The Bill of Materials (BOM) for your ZebOS-XP software, including information about the operating system, protocols, directory structure, and files.

Configure the Build Environment

There are two ways to configure the build environment:

- Automatically by running the `config.sh` script provided by IP Infusion Inc.
- Manually by setting each build option individually

Configure the Build Environment Automatically

The ZebOS-XP configuration script automatically detects most host configurations. Perform the following steps to configure the build environment. Running the `configure` process takes some time. While running, the script generates messages explaining which features it is checking.

1. Verify that all enabled and disabled configuration options match your requirements.
2. Change to the directory that contains the package's source code:
   ```
   cd /myZebOSpath/ZebOS
   ```
3. Create a symbolic link to the Linux kernel source which is used by the `setup.sh` script. This path is set up when you install the `linux-headers-*.deb` package as described in Appendix C, *Install and Build Debian 8*:
   ```
   ln -sf /usr/src/linux-headers-3.16.7-xp-1.4.0.0 linux-source
   ```
4. Source the `setup.sh` script provided by IP Infusion Inc. which sets shell variables:
   ```
   source setup.sh
   ```
   - For 64 bit SMP or non SMP Debian OS, run the following step also:
     ```
     ln -sf ${KERNEL_SOURCE}/include/uapi/linux/{mroute.h,mroute6.h}header/linux
     ```
5. Run the `config.sh` script provided by IP Infusion Inc.:
   ```
   config.sh
   ```
   This script executes the `configure` command and enables or disables configuration options depending on the ZebOS-XP modules purchased.

Configure Build Environment Manually

To set configuration options manually, review the `config.sh` script that came with your ZebOS-XP software before proceeding. Remember, some configuration options are already enabled, depending on the software you have purchased.

1. Change to the directory containing the ZebOS-XP source code:
   ```
   cd /myZebOSpath/ZebOS
   ```
2. Create a symbolic link to the Linux kernel source which is used by the `setup.sh` script:
   ```
   ln -sf /usr/src/linux-headers-3.16.7-xp-1.4.0.0 linux-source
   ```

3. Source the `setup.sh` script provided by IP Infusion Inc. which sets shell variables:

   ```
   source setup.sh
   ```

   - For 64 bit SMP or non SMP Debian OS, run the following step also:

     ```
     ln -sf ${KERNEL_SOURCE}/include/uapi/linux/{mroute.h,mroute6.h}header/linux
     ```

4. Choose the configure options to disable. Refer to the `config.sh` script to see the options enabled by default.

5. Choose the configuration options to enable from the tables in Appendix A, *Build Configuration Options*.

   Some options have dependencies, so if all the required options are not enabled for an affected module, you might get an error when you run the `make` command.

   For example, if you have enabled the OSPFv6 option, but did not enable the IPv6 option, then you run the `make` command, you might see:

   ```
   make ospf6d nsm
   Make aborted: Unable to find ../../ospf6d/Makefile
   make: *** [ospf6d] Error 1
   ```

6. Configure the package for the system:

   ```
   ./configure <options>
   ```

   where `<options>` are any build configuration options that you have chosen to enable or disable, for example:

   ```
   --enable-nsm --disable-ipv6
   ```

Often-Used Options

This section describes often-used configuration options.

Layer-2 Software Forwarder

Use the following configuration options to enable the Software Forwarder provided by IP Infusion Inc.:

```
--enable-hal
--enable-swfwdr
```

MPLS Forwarder

Use the following configuration option to enable the MPLS Forwarder provided by IP Infusion Inc.:

```
--enable-mpls-fwd
```

Virtual Private Network (VPN)

To build the VPN feature and enable VRF, enable the following options while compiling:

`--enable-vrf`

`--enable-mpls-fwd` Applicable only when using the ZebOS-XP MPLS Forwarder

Note: To establish a Virtual Private Network, the BGP, OSPF (or RIP), LDP, and NSM daemons must be running.

Virtual Routing (VR)

For Virtual Routing, enable the following configuration option while compiling:

```
--enable-vr
```

Note: Enabling VR automatically enables VRF (Virtual Routing Forwarding) support for the Network Services Manager (NSM), and VR support for LIB, NSM, OSPF and BGP.

Set Up the System for Virtual Routing

Ensure that your system is installed with all of the Advanced Routing options turned on. For testing purposes, check to see that the IPROUTE2 utility package is installed.

High Availability (HA)

To enable HA, refer to the information below.

- For High Availability features, enable the following option:

  ```
  --enable-opensaf
  ```
- To use the Fault Management Simulation, enable the following option:

  ```
  --enable-fm-sim
  ```
- To run HA in debugging mode, enable the following option:

  ```
  --enable-zebos-debug
  ```

Compilation Requirements

To build ZebOS-XP, you need the following versions of Linux development tools:

- `autoconf` version 2.69
- `gcc` version 4.9.2
- `binutils` version 2.25
- `libc6` version 2.19

Compile ZebOS-XP

After configuring the build environment, compile ZebOS-XP.

1. Change to the directory containing the ZebOS-XP source code:

   ```
   cd platform/linux
   ```
2. Run the `make clean` utility:

   ```
   make clean
   ```
3. Run the `make` utility. Run `make all` to compile all available modules, or just compile specific protocols by specifying the protocol name. Note that `make install` implies `make all`.

   For example:

   `make <protocol>|all` Makes all modules for the selected `<protocol>`
   `make bgpd` Makes the BGP modules
   `make install` Makes all modules and daemons
4. If you choose to make individual modules, and do not run the `make all` or `make install` command, and you need the MPLS Forwarder, run the `make all-mpls` command to build it.
5. Likewise if you choose to make individual modules and you need Layer 2 support, run `make all-layer2` to make the Layer 2 support modules.

   Note: The `make` command displays available make targets. You can make individual targets in separate directories by entering `make target-directory`.

   For example:

   `make all-pal` makes all PAL-related files in the PAL target directory
   `make all-lib` makes all library files in the LIB directory
   `make nsm` makes the NSM daemon

Table 2-2 lists the `makefile` commands and their application.

Table 2-2: Makefile commands

Make Targets	Function	Comments
`distclean`	Removes all working files not in the distribution	Applies only at the top level
`install`	Makes and Installs binaries and samples	Applies only at the top level
`all`	Makes all available modules	Applies to the top level and to all available modules
`dep`	Makes all dependencies	Applies to the top level and to all available modules
`forcedep`	Forces all dependencies to be remade	Applies to the top level and to all available modules
`clean`	Cleans up from a previous make	Applies to the top level and to all available modules
<protocol>	Makes the protocol daemon(s) for ZebOS-XP modules	`authd, bgpd, elmid, gvrpd, isisd, isis6d, lagd, ldpd, lmpd, mstpd, nsm, oamd, onmd, ospfd, ospf6d, pimd, pim6d, ripd, ripngd, rstpd, rsvpd, stpd`

Configure Host and Target for Cross-Compiling

Use the `configure` command with the following options when cross-compiling. For example:

```
./configure --host=<xxx> --build=<xxx> --target=<xxx>
```

where <xxx> represents the names of the host, build and target machines, for example:

```
./configure \
--host=i686-redhat-linux-gnu \
--target=powerpc-wrs-linux-gnu \
--build=powerpc-wrs-linux-gnu
```

Build Loadable Kernel Modules

Now that ZebOS-XP is installed and configured, specific kernel modules must be built. Follow the steps in the two sections that follow.

Build Layer 2 Kernel Modules

1. Enable the following configuration options to enable the Software Forwarder provided by IP Infusion Inc.:

   ```
   --enable-hal
   --enable-swfwdr
   ```

2. Change to the `platform` directory:

   ```
   cd platform/linux
   ```

3. Run the following `make` utility to build the Layer 2 module:

   ```
   make all-layer2
   ```

   This utility builds a `layer2_module.ko` file in the `platform/linux/bin` directory.

4. Install the Layer 2 module using the appropriate command for your environment in the `platform/linux/bin` directory:

   ```
   insmod layer2_module.ko
   ```

5. Execute the appropriate command for your environment to ensure that the Layer 2 kernel module is gracefully removed in the event of a reboot or shutdown:

   ```
   rmmod layer2_module.ko
   ```

Build MPLS Kernel Modules

1. Enable the following configuration option to enable the MPLS Forwarder provided by IP Infusion Inc.:

   ```
   --enable-mpls-fwd
   ```

2. Change to the `platform` directory:

   ```
   cd platform/linux
   ```

3. Run the `make` utility to build the MPLS module:

   `make all` (or)

   `make all-mpls` Use this command if you do not want to run `make all`.

   This builds the or the `mpls_module.ko` file in the `platform/linux/bin` directory.

4. Install the MPLS module using the appropriate command for your environment in the `platform/linux/bin` directory:

   ```
   insmod mpls_module.ko
   ```

5. Execute the appropriate command for your environment to ensure that the MPLS Forwarder kernel module is gracefully removed in the event of a reboot or shutdown:

   ```
   rmmod mpls_module.ko
   ```

Appendix A Build Configuration Options

The tables in this appendix show the ZebOS-XP configuration options that you set to configure the build environment. Each option is listed by protocol or service group. Examples of protocols are BGP or RIP; examples of service groups are Layer 2 or Multicast.

Note: The notation “Default” means that this is the default setting for the option.The default settings are specified in the `configure.ac` script and apply the most common feature set of ZebOS-XP.

BFD

Option flag	Description
--enable-bfd	Enable BFD (Default)
--disable-bfd	Disable BFD

BGP

Option flag	Description
--enable-bgpd	Build the BGP daemon (Default)
--disable-bgpd	Do not build BGP daemon
--enable-bgp-dump	Enable BGP dump features
--disable-bgp-dump	Disable BGP dump features (Default)
--enable-bgpd-extasn	Enable BGP extended ASN capability (Default)
--disable-bgpd-extasn	Disable BGP extended ASN capability
--enable-bgp-rt2fib	Enable BGP routes to be added to or deleted from NSM and FIB (Default)
--disable-bgp-rt2fib	Disable BGP routes from being added to or deleted from NSM and FIB
--enable-order-nexthop	Enable BGP Order NextHop
--disable-order-nexthop	Disable BGP Order NextHop (Default)
--enable-bgp-sfd	Enable Selective Route Download (SFD)
--disable-bgp-sfd	Disable Selective Route Download (SFD) (Default)

Data Center

Option flag	Description
--enable-dcb	Enable Data Center Bridging
--disable-dcb	Disable Data Center Bridging (Default)
--enable-ecpd	Enable Edge Control Protocol (ECP) (Default)
--disable-ecpd	Disable Edge Control Protocol
--enable-spbd	Enable Shortest Path Bridging (SPB) (requires provider-bridge to be enabled)
--disable-spbd	Disable Shortest Path Bridging (Default)
--enable-trilld	Enable TRILL
--disable-trilld	Disable TRILL (Default)

High Availability

Option Flag	Description
--enable-ha	Enable High Availability features
--disable-ha	Do not enable the High Availability features (Default)
--enable-fm-sim	Enable the Fault Management Simulation
--disable-fm-sim	Do not enable the Fault Management Simulation (Default)
--enable-open-saf	Enable support for OpenSAF checkpointing
--disable-open-saf	Disable support for OpenSAF checkpointing (Default)
--enable-zebos-debug	Enable debugging for High Availability features
--disable-zebos-debug	Do not enable debugging for High Availability features (Default)
--enable-ha-ospf-gr	Enable High Availability support for OSPF resiliency using Graceful Restart feature
--disable-ha-ospf-gr	Disable High Availability support for OSPF using Graceful Restart (Default, works in SSO mode)

Hardware Integration

Option flag	Description
--enable-broadcom[=bmw|gto]	Build the Broadcom Hardware Services Layer; optional parameters are `bmw` or `gto`
--enable-marvell	Build the Marvell Hardware Services Layer
--enable-wintegra	Build the Wintegra Hardware Services Layer
--enable-hal	Build the Hardware Abstraction Layer (HAL)
--enable-user-hsl	Build the User Hardware Services Layer (HSL)
--enable-ccm-hw-offload	Offload CCM to hardware for Wintegra WinPath3
--enable-Y1731-hw-offload	Offload Y1731 to hardware Wintegra WinPath3

IMI

Option flag	Description
--disable-acl	Disable ACL for Basic Access
--enable-basic-access	Enable the Basic Access features (DHCP, DNS, NAT and ACL)
--enable-config-compare	Enables the feature for displaying an asterisk (*) at the head of the CLI prompt when a change is made to the configuration without saving.
--disable-config-compare	Disables the feature for displaying the asterisk (*) at the head of the CLI prompt when a change is made to the configuration without saving
--enable-dhcp-client	Enable the DHCP client for IMI
--disable-dhcp-client	Disable the DHCP client for IMI (Default)
--enable-dhcp-server	Enable the DHCP server for IMI
--disable-dhcp-server	Disable the DHCP server for IMI (Default)
--enable-dhcp-update-style	Enable the DHCP server update style
--disable-dhcp-update-style	Disable the DHCP server update style (Default)
--enable-dns-client	Enable the DNS client for IMI
--disable-dns-client	Disable the DNS client for IMI (Default)
--enable-imi	Enable the Integrated Management Interface (IMI) Note: Enabling IMI does not enable the IMI Shell (IMISH) automatically.

Option flag	Description
--disable-imi	Disable the Integrated Management Interface (Default)
--enable-imish	Enable the IMI Shell (IMISH)
--disable-imish	Disable IMI Shell (Default)
--enable-imi-sysconfig	Enable Read system configurations on startup
--disable-imi-sysconfig	Disable Read system configurations on startup (Default)
--enable-multi-conf-ses	Enable multiple configuration sessions, to allow more than one user to enter Configure mode Note: This option requires IMI Shell.
--disable-multi-conf-ses	Disable multiple configuration sessions (Default)
--enable-nat	Enable Network Address Translation for IMI
--disable-nat	Disable Network Address Translation for IMI (Default)
--disable-ntp	Disable Network Time Protocol
--enable-pppoe-client	Enable the PPPoE client for IMI
--disable-pppoe-client	Disable the PPPoE client for IMI (Default)
--enable-vlogd	Enable VR context-sensitive login
--disable-vlogd	Disable VR context-sensitive login (Default)

IPv6

Option flag	Description
--enable-ipv6	Enable IPv6-related features and daemons (Default)
--disable-ipv6	Disable IPv6-related features and daemons
--enable-ipv6-transit	Enable IPv4-to-IPv6 transition
--disable-ipv6-transit	Disable IPv4-to-IPv6 transition
--enable-ndd	Enable neighbor discovery (Default)
--disable-ndd	Disable neighbor discovery

IS-IS

Option flag	Description
--enable-isisd	Build the IS-IS daemon (Default Linux)
--disable-isisd	Do not build the IS-IS daemon
--enable-isis6d	Build the IS-IS6 daemon (Default Linux)
--disable-isis6d	Do not build the IS-IS6 daemon
--enable-multi-topology	Enable IS-IS multi-topology support
--disable-multi-topology	Do not enable IS-IS multi-topology support.(Default)

LDP

Option flag	Description
--enable-ldpd	Build the LDP daemon (Default)
--disable-ldpd	Do not build the LDP daemon
--enable-ldp-suppress-fec	Enable Suppress LDP FEC (Default)
--disable-ldp-suppress-fec	Disable Suppress LDP FEC

Layer 2

Option flag	Description
--enable-authd	Build the 802.1x authentication module daemon
--disable-authd	Do not build the 802.1x authentication module daemon (Default)
--enable-b-pbb	Enable Provider Backbone Bridging (PBB) with B-tag
--disable-b-pbb	Disable Provider Backbone Bridging (PBB) with B-tag (Default)
--enable-cfm-Y1731	Enable Connectivity Fault Management (CFM) Y.1731 messaging extensions
--disable-cfm-Y1731	Disable Connectivity Fault Management (CFM) Y.1731 messaging extensions (Default)
--enable-g8031	Enable Ethernet Linear protection
--enable-cfm-cli-v2	Enable Connectivity Fault Management (CFM) version 2

Option flag	Description
--disable-cfm-cli-v2	Disable Connectivity Fault Management (CFM) version 2 (Default)
--enable-default-bridge	Creates a default bridge during start up; not supported on Broadcom platforms
--enable-drni	Enable Distributed Resilient Network Interconnect (DRNI) (Default)
--disable-drni	Disable DRNI
--enable-elmid	Build the Ethernet Link Management Interface (ELMI) daemon
--disable-elmid	Do not build the Ethernet Link Management Interface daemon (Default)
--disable-fm-smi	Disable fault management simple management interface
--enable-g8031	Enable Ethernet Linear Protection
--disable-g8031	Disable Ethernet Linear protection (Default)
--enable-g8032=v1	Enable Ethernet Ring protection (ITU-T Rec. G.8032 2008)
--enable-g8032 --enable-g8032=v2	Enable Ethernet Ring protection (ITU-T Rec. G.8032 2012)
--disable-g8032	Disable Ethernet Ring Protection (Default)
--enable-gmrp	Enable the GARP Multicast Registration Protocol (GMRP) feature
--disable-gmrp	Do not enable the GMRP feature (Default)
--enable-gvrp	Enable the GARP VLAN Registration Protocol (GVRP) feature
--disable-gvrp	Do not enable the GVRP feature (Default)
--enable-intervlan-routing	Enables the use of inter-VLAN routing
--enable-hqos	Enable Hierarchical Quality of Service
--disable-hqos	Disable Hierarchical Quality of Service (Default)
--disable-intervlan-routing	Disables the use of inter-VLAN routing
--enable-i-pbb	Enable Provider Backbone Bridging (PBB) with I-tag.
--disable-i-pbb	Disable Provider Backbone Bridging (PBB) with I-tag (Default)
--enable-ipi-drni	Enable ZebOS-XP extensions for DRNI (Default)
--disable-ipi-drni	Disable ZebOS-XP extensions for DRNI
--enable-lacpd	Enable Link Aggregation Control Protocol (LACP) (Default)
--disable-lacpd	Disable LACP
--enable-lacpv2	Enable LACP version 2 (Default)
--disable-lacpv2	Disable LACP version 2
--enable-mcec	Enable Multichassis EtherChannel (MCEC)

Option flag	Description
--disable-mcec	Disable Multichassis EtherChannel (MCEC) (Default)
--enable-mmrp	Enable Multiple Multicast Registration Protocol (MMRP)
--disable-mmrp	Disable MMRP (Default)
--enable-mstpd	Build the Multiple Spanning Tree Protocol (MSTP) daemon
--disable-mstpd	Do not build the MSTP daemon (Default)
--enable-mvrp	Enable the Multiple VLAN Registration Protocol (MVRP) features
--disable-mvrp	Disable the MVRP features (Default)
--enable-onmd	Build the operations, administration and management (ONM) daemon
--disable-onmd	Do not build the ONM daemon
--disable-qos	Disable Layer 2 Quality of Service module, traffic shaping, and policing
--enable-provider-bridge	Enable provider bridge
--disable-provider-bridge	Disable provider bridge
--enable-ptpd	Enable Precision Time Protocol (PTP) (Default)
--disable-ptpd	Disable Precision Time Protocol (PTP)
--enable-rate-limit	Enable rate limiting (broadcast storm control)
--disable-rate-limit	Disable the rate limiting feature
--enable-rmond	Build the Remote Monitoring (RMON) daemon
--disable-rmond	Do not build the RMON daemon
--enable-rpvst-plus	Enable rapid per-VLAN Spanning Tree
--enable-rpvst-plus	Disable rapid per-VLAN Spanning Tree (Default)
--enable-rstpd	Build the Rapid Spanning Tree Protocol (RSTP) daemon
--disable-rstpd	Do not build the RSTP daemon (Default)
--disable-smi	Disable simple management interface (SMI)
--enable-advance-mirror	Enable Switched Port Analyzer (SPAN)
--disable-advance-mirror	Disable Switched Port Analyzer (SPAN) (Default)
--enable-stpd	Build the Spanning Tree Protocol (STP) daemon
--disable-stpd	Do not build the STP daemon (Default)
--enable-swfwdr	Enable the Layer 2 software forwarder
--disable-swfwdr	Disable the Layer 2 software forwarder (Default)
--enable-synced	Enable Synchronous Ethernet (SynchE)

Option flag	Description
--disable-synced	Disable Synchronous Ethernet (Default)
--enable-udld	Enable Unidirectional Link Detection (UDLD)
--disable-udld	Disable Unidirectional Link Detection (UDLD) (Default)
--enable-vlan	Enable VLAN-aware bridging with either STP, RSTP, or MSTP
--disable-vlan	Disable VLAN-aware bridging
--enable-vlan-class	Enable VLAN classification (only available for Broadcom integrations)
--disable-vlan-class	Disable VLAN classification
--enable-vlan-stacking	Enable VLAN stacking (only available for Broadcom integrations)
--disable-vlan-stacking	Disable VLAN stacking
--enable-vxlan	Enable Virtual eXtensible Local Area Network (VXLAN)
--disable-vxlan	Disable Virtual eXtensible Local Area Network (VXLAN) (Default)

Memory Manager

Option flag	Description
--enable-memmgr	Enable the use of all allocation methods for various `mtypes`
--disable-memmgr	Disable allows only the `heap` method (Default)

MPLS

Option flag	Description
--enable-6pe	Enable IPv6 Provider Edge (6PE) router support.
--disable-6pe	Disable 6PE router support. (Default)
--enable-bgp-vpls	Enable BGP VPLS
--disable-bgp-vpls	Disable BGP VPLS (Default)
--enable-diffserv	Enable MPLS DiffServ (Differential Services) support
--disable-diffserv	Disable MPLS DiffServ support (Default)
--enable-dste	Enable MPLS DiffServ-TE (Traffic Engineering) support.

Option flag	Description
--disable-dste	Disable MPLS DiffServ-TE support (Default)
--enable-fwdr-stub	Enable this option to test mpls control plane protocol scalability (LDP/RSVP-TE). Note: This will stub L2 and MPLS Linux kernel calls.By default, this option is disabled.
--enable-mpls	Enable the Multi-Protocol Label Switching (MPLS) feature. This option is automatically turned on if any MPLS-specific options (LDP, RSVP, MPLS-VC, VPLS, MPLS Forwarder, VRF, CSPF and TE) are selected.
--disable-mpls	Disable the MPLS supported in the NSM (Default) Note: Do not disable MPLS with this option if any MPLS options are selected.
--enable-mpls-atm	Enable MPLS ATM Note: Contact IP Infusion Inc. for this feature's availability. MPLS and MPLS VC should be enabled before enabling MPLS ATM, otherwise, MPLS ATM will automatically enable MPLS and MPLS VC.
--disable-mpls-atm	Disable MPLS ATM (Default)
--enable-mpls-bk-lsp	Enable MPLS to send all primary, backup, and secondary LSPs to the MPLS Forwarder Note: Hardware integration and switch-over in the MPLS Forwarder is not supported.
--disable-mpls-bk-lsp	Disable MPLS from sending all primary, backup, and secondary LSPs to the MPLS Forwarder (Default)
--enable-mpls-frr	Enable the Fast Reroute feature of RSVP-TE
--disable-mpls-frr	Disable the Fast Reroute feature of RSVP-TE (Default)
--enable-mpls-fwd	Enable the Multi Protocol Label Switching (MPLS) Forwarder support
--disable-mpls-fwd	Disable the MPLS Forwarder support (Default)
--enable-mpls-oam	Enable the MPLS Operations and Management (OAM) features
--disable-mpls-oam	Disable the MPLS OAM features (Default)
--enable-mpls-p2mp	Enable MPLS point-to-multipoint LSPs
--disable-mpls-p2mp	Disable MPLS point-to-multipoint LSPs (Default)
--enable-mpls-tdm	Enable MPLS time-division multiplexing (TDM) Note: Contact IP Infusion Inc. for this feature's availability. MPLS and MPLS VC should be enabled before enabling MPLS TDM, otherwise, MPLS TDM will automatically enable MPLS and MPLS VC.
--disable-mpls-tdm	Disable MPLS time-division multiplexing (TDM) (Default)
--enable-ietf-mpls-tp-oam	Enable IETF MPLS TP
--enable-itut-mpls-tp-oam	Enable ITU-T MPLS TP

Option flag	Description
--disable-ietf-mpls-tp-oam	Disable IETF MPLS TP
--disable-itut-mpls-tp-oam	Disable ITU-T MPLS TP
--enable-mpls-vc	Enable the MPLS-based Virtual Circuit feature
--disable-mpls-vc	Disable the MPLS-based Virtual Circuit feature (Default)
--enable-mpls-tdm-vc	Enable SAToP (Structure Agnostic TDM over Packet). Only enable this option if --enable-mpls-vc is also set.
--disable-mpls-tdm-vc	Disable SAToP (Default)
--enable-ms-pw	Enable Multi-Segment Pseudowire (MS-PW) support
--disable-ms-pw	Disable MS-PW support (Default)
--enable-vpls	Enable the MPLS-based Virtual Private LAN Service (VPLS) feature
--enable-sr-mpls	Enable segment routing
--disable-sr-mpls	Disable segment routing (Default)
--disable-vpls	Disable the MPLS-based VPLS feature (Default)
--enable-vccv	Enable Virtual Circuit Connectivity Verification (VCCV)
--disable-vccv	Disable VCCV (Default)

Multicast

Option flag	Description
--enable-igmp-snoop	Enable Internet Group Management Protocol (IGMP) snooping
--disable-igmp-snoop	Disable IGMP snooping (Default)
--enable-igmp-v3	Enable IGMPv3 (Default)
--disable-igmp-v3	Disable IGMPv3. This option disables only IGMPv3, but not IGMPv1 or IGMPv2.
--enable-ipv4	This option is replaced by enable-mcast-ipv4
--enable-mcast4-intf-limit	Enable IPv4 multicast interface limit check (Default)
--disable-mcast4-intf-limit	Disable IPv4 multicast interface limit check
--enable-mcast4-intf-limit-ignore	Enable ignore IPv4 multicast interface limit check for Control Plane verification
--disable-mcast4-intf-limit-ignore	Disable ignore IPv4 multicast interface limit check for Control Plane verification (Default)

Option flag	Description
--enable-mcast6-intf-limit	Enable IPv6 multicast interface limit check (Default)
--disable-mcast6-intf-limit	Disable IPv6 multicast interface limit check
--enable-mcast6-intf-limit-ignore	Enable ignore IPv6 multicast interface limit check for Control Plane verification
--disable-mcast6-intf-limit-ignore	Disable ignore IPv6 multicast interface limit check for Control Plane verification (Default)
--enable-mcast-ipv4	Enable IPv4 in the multicast daemons (mribd and pimfd)
--disable-mcast-ipv4	Disable IPv4 in the multicast daemons (mribd and pimfd) (Default)
--enable-mcast-ipv6	Enable IPv6 in the multicast daemons (mribd and pimfd)
--disable-mcast-ipv6	Disable IPv6 in the multicast daemons (mribd and pimfd)
--enable-mld-snoop	Enable Multicast Listener Discovery (MLD) snooping
--disable-mld-snoop	Disable MLD snooping (Default)
--enable-mribd	Enable mribd (Default)
--disable-mribd	Disable mribd
--enable-pimd	Build the PIM-SM daemon (Default)
--disable-pimd	Do not build the PIM-SM daemon
--enable-pim-bidir	Enable bidirectional PIM
--disable-pim-bidir	Disable bidirectional PIM (Default)
--enable-pim-ipv4	Enable PIMv4 under PIM daemon; requires mcast-ipv4 to be enabled
--enable-pim-ecmp-redirect	Enable PIM ECMP redirect
--disable-pim-ecmp-redirect	Disable PIM ECMP redirect (Default)
--enable-pim-ipv6	Enable PIMv6 under PIM daemon; requires mcast-ipv6 to be enabled
--enable-pim-sm	Enable PIM-SM feature under the PIM daemon
--enable-pim-dm	Enable the PIM-DM feature under the PIM daemon
--enable-pim-ssm	Enable PIM Source-Specific Multicast
--enable-pim-msdp	Enable PIM MSDP
--enable-pim-msdp-api	Enable PIM MSDP API feature inside PIMd
--enable-pim-smdm	Enable PIM sparse-dense mode

Network Services Module

Option flag	Description
--enable-agentx	Enable AgentX protocol
--disable-agentx	Disable AgentX protocol (Default)
--enable-if-arbiter	Enable interface arbiter to find interfaces recently added to the kernel and add them to ZebOS-XP
--disable-if-arbiter	Disable if-arbiter (Default)
--enable-if-unnumbered	Enable the NSM IPv4/IPv6 unnumbered interface feature
--disable-if-unnumbered	Disable the NSM IPv4/IPv6 unnumbered interface feature (Default)
--enable-ipv4-tunnel	Enable IPv4 tunnelling
--enable-ipsec	Enable Internet Protocol Security
--disable-ipsec	Disable Internet Protocol Security (Default)
--disable-ipv4-tunnel	Disable IPv4 tunnelling (Default)
--enable-multiple-fib	Enable the Multiple FIB kernel support (Default)
--disable-multiple-fib	Disable the Multiple FIB kernel support
--enable-netlink	Enable GNU/Linux-NetLink interface. The ZebOS-XP configure script detects the NetLink interface by checking the header file. When the header file does not match to the current running kernel, the configure script will not turn on NetLink support.
--disable-netlink	Disable the NetLink interface (Default)
--enable-nsm	Build the NSM daemon (Default)
--disable-nsm	Do not build the NSM daemon
--enable-rtadv	Enable the Router Advertisement feature (Default)
--disable-rtadv	Disable the Router Advertisement feature
--enable-snmp	Enable SNMP support for: BGP, IS-IS, LDP, NSM, OSPF, PIM, RIP (Default)
--disable-snmp	Disable SNMP support
--enable-tcp-message	Enable TCP/IP socket connection between daemons
--disable-tcp-message	Disable TCP/IP socket connection between daemons (Default)

OSPF

Option flag	Description
--enable-ospf6-od	Enable hello suppression and LSA suppression (“on demand”)
--disable-ospf6-od	Disable hello suppression and LSA suppression (“on demand”)
--enable-ospf-multi-area	Enable support for multiple OSPF areas on the same IP interface
--disable-ospf-multi-area	Disable support for multiple OSPF areas on the same IP interface (Default)
--enable-ospf-multi-inst	Enable support for multiple OSPF instances on the same IP interface Note: Enabling this flag does not automatically enable support for this feature: this feature also requires using the enable-ext-ospf-multi-inst command in Configure mode.
--disable-ospf-multi-inst	Disable support for multiple OSPF instances on the same IP interface (Default)
--enable-ospf-db-overflow	Limit the maximum number of LSAs
--disable-ospf-db-overflow	Do not limit the maximum number of LSAs
--enable-ospfd	Build the OSPF daemon (Default)
--disable-ospfd	Do not build the OSPF daemon
--enable-ospf6d	Build the OSPF6 daemon (Default)
--disable-ospf6d	Do not build the OSPF6 daemon
--enable-pece-ospf	Adds support for `draft-ishiguro-ppvpn-pe-ce-ospf-01.txt`, which specifies the method of using multiple instances of OSPF on the PE to CE link to support multiple customer networks, when provisioning BGP-MPLS VPNs
--disable-pece-ospf	Disable support for `draft-ishiguro-ppvpn-pe-ce-ospf-01.txt` (Default)
--enable-pece-ospf6	Enable OSPFv3 as the CE/PE protocol to support VPN
--disable-pece-ospf6	Disable OSPFv3 as the CE/PE protocol to support VPN (Default)
--enable-ext-ospf-multi-inst	Enable support for extended multiple OSPF instances on the same IP interface Note: Enabling this flag does not automatically enable support for this feature: this feature also requires using the enable-ext-ospf-multi-inst command in Configure mode.
--disable-ext-ospf-multi-inst	Disable support for extended multiple OSPF instances on the same IP interface (Default)
--enable-ospf-lfa	Enable loop-free alternate/fast reroute
--disable-ospf-lfa	Disable loop-free alternate/fast reroute

RIP

Option flag	Description
--enable-pece-rip	Enable RIP as the CE/PE protocol to support VPN (Virtual Private Network)
--disable-pece-rip	Disable RIP as the CE/PE protocol (Default)
--enable-pece-ripng	Enable RIPNG as the CE/PE protocol to support VPN
--disable-pece-ripng	Disable RIPNG as the CE/PE protocol to support VPN (Default)
--enable-ripd	Build the RIP daemon (Default)
--disable-ripd	Do not build RIP daemon
--enable-ripngd	Build the RIPng daemon (Default)
--disable-ripngd	Do not build RIPng daemon

System Management

Option flag	Description
--enable-hostpd	Enable all system management options
--disable-hostpd	Disable system management (Default)
--enable-hostp-aaa	Enable AAA
--disable-hostp-aaa	Disable AAA (Default)
--enable-hostp-dhcp-relay	Enable DHCP relay
--disable-hostp-dhcp-relay	Disable DHCP relay (Default)
--enable-hostp-ldap-client	Enable LDAP client
--disable-hostp-ldap-client	Disable LDAP client (Default)
--enable-hostp-ntp	Enable NTP
--disable-hostp-ntp	Disable NTP (Default)
--enable-hostp-radius-client	Enable RADIUS client
--disable-hostp-radius-client	Disable RADIUS client (Default)
--enable-hostp-remote_mgmt	Enable remote managementt
--disable-hostp-remote_mgmt	Disable remote management (Default)

Option flag	Description
--enable-hostp-role-mgmt	Enable role management
--disable-hostp-remote_mgmt	Disable role management (Default)
--enable-hostp-ssh-client	Enable SSH client
--disable-hostp-ssh-client	Disable SSH client (Default)
--enable-hostp-ssh-server	Enable SSH server
--disable-hostp-ssh-server	Disable SSH server (Default)
--enable-hostp-syslog	Enable syslog
--enable-hostp-tacacas-client	Enable TACACAS client
--disable-hostp-tacacas-client	Disable TACACAS client (Default)
--enable-hostp-telnet-client	Enable Telnet client
--disable-hostp-telnet-client	Disable Telnet client (Default)
--enable-hostp-telnet-server	Enable Telnet server
--disable-hostp-telnet-server	Disable Telnet server (Default)
--enable-hostp-user-mgmt	Enable user management
--disable-hostp-user-mgmt	Disable user management (Default)

System Options

Option flag	Description
--enable-bigendian	Enable Big Endian support
--disable-bigendian	Disable Big Endian support
--enable-hostname-change	Enable the ability to change the host name using the `hostname` command
--disable-hostname-change	Disable the host name change feature (Default)
--enable-restart	Enable Graceful and Hitless Restart features
--disable-restart	Disable Graceful and Hitless Restart features (Default)
--enable-static	Enable static library linking. Prevents linking with the shared libraries on systems that support dynamic linking
--disable-static	Disable static library linking (Default)
--enable-storage-dev	Enable Storage Device support (Default)

Option flag	Description
--disable-storage-dev	Disable Storage Device support
--with-ipnet2	Use IPNet for IPv6 support
--with-pne37	Build with PNS 3.7 (VxWorks 6.7)
--enable-poe	Enable Power over Ethernet (PoE)
--disable-poe	Disable Power over Ethernet (PoE) (Default)
--enable-pbr	Enable Policy-Based Routing (PBR)
--disable-pbr	Disable PBR (Default)
--enable-ribd	Enable unicast Routing Information Base (Default)
--disable-ribd	Disable unicast Routing Information Base
--enable-lfa	Enable loop-free alternate/fast reroute for `ribd`
--disable-lfa	Disable loop-free alternate/fast reroute for `ribd`
--enable-ofdpa	Enable Broadcom OpenFlow Data Plane Abstraction (OF-DPA) (Default)
--disable-ofdpa	Disable Broadcom OpenFlow Data Plane Abstraction (OF-DPA)
--enable-ofld	Enable OpenFlow
--disable-ofld	Disable OpenFlow (Default)
--enable-sflow	Enable sampling flow
--disable-sflow	Disable sampling flow (Default)
--enable-tfo	Enable Trigger Failover (TFO)
--disable-tfo	Disable Trigger Failover (TFO) (Default)

Traffic Engineering

Option flag	Description
--enable-isis-cspf	Enable the CSPF (Constrained Shortest Path First) feature and all TE options for IS-IS
--disable-isis-cspf	Disable the CSPF feature and all TE options for IS-IS (Default)
--enable-ospf-cspf	Enable the CSPF feature and all TE options for OSPF
--disable-ospf-cspf	Disable the CSPF feature and all TE options for OSPF (Default)
--enable-ospf6-cspf	Enable the CSPF feature and all TE options for OSPFv3
--disable-ospf6-cspf	Disable the CSPF feature and all TE options for OSPFv3 (Default)

Option flag	Description
--enable-pbb-te	Enable Provider Backbone Bridge-Traffic Engineering (PBB-TE)
--disable-pbb-te	Disable PBB-TE (Default)
--enable-rsvp-graceful	Enable the RSVP Graceful Restart feature
--disable-rsvp-graceful	Disable the RSVP Graceful Restart feature (Default)
--enable-rsvpd	Build the RSVP daemon (Default)
--disable-rsvpd	Do not build the RSVP daemon
--enable-te	Enable TE for IS-IS, OSPF, RSVP, and QoS Module Stub
--disable-te	Disable TE support (Default)

Virtual Routing/Forwarding

Option flag	Description
--enable-vr	Enable Virtual Routing
--disable-vr	Disable VR (Default)
--enable-vrf	Enable Virtual Router Forwarding (VRF) Note: If you have enabled VR, you do not need to enable VRF. Enabling VR automatically enables VRF.
--disable-vrf	Disable VRF (Default)
--enable-vrf-ns	Enable VRF namespace
--disable-vrf-ns	Disable VRF namespace (Default)

VRRP

Option flag	Description
--enable-vrrpd	Enable Virtual Router Redundancy Protocol daemon Note: When used alone, this enables the RFC 3678-compliant version.
--disable-vrrpd	Disable vrrpd (Default)
--enable-vrrp-vmac	Enable VRRP Virtual MAC address (Default). Use this option to enable virtual MAC addressing for single groups.

Option flag	Description
--disable-vrrp-vmac	Disable VRRP Virtual MAC address. Use this option to disable virtual MAC addressing and use physical MAC addressing for multiple groups.
--disable-vrrp-link-addr	Allows virtual IP as a connected address
--enable-vrrp-v3	Enable VRRP V3 (RFC 5798); VRRP v3 is enabled by default when you enable VRRP using `--enable-vrrpd`. VRRP v3 can be disabled using `--disable-vrrp-v3`. If `--enable-vrrpd` is not given, VRRP v3 is also disabled by default
--disable-vrrp-v3	Disable VRRP V3 and enable VRRP v2 (Default)

Appendix B Running ZebOS-XP Daemons

This appendix has instructions for starting, stopping, and configuring ZebOS-XP daemons. A table lists the daemons, their configuration files, and their port numbers.

Note: Daemons may be enabled only for those modules that are licensed and purchased.

ZebOS-XP Daemons

Daemons manage the modules that make up ZebOS-XP. Table 1 lists each daemon and describes what it manages.

Table 1: ZebOS-XP daemons

Daemon name	Manages	Default configuration file	Sample file	Port Number
`authd`	802.1x Authentication	`authd.conf`	`authd.conf.sample`	`2613/tcp`
`bgpd`	Border Gateway Protocol (BGP4/ BGP4+)	`bgpd.conf`	`bgpd.conf.sample`	`2605/tcp`
`ecpd`	Edge Control Protocol (ECP)	`ecpd.conf`	N/A	2665/tcp
`elmid`	Ethernet Link Management Interface (E-LMI)	`elmid.conf`	N/A	`2653/tcp`
`hostpd`	Host protocols	`hostpd.conf`	N/A	`2663/tcp`
`imi`	Integrated Management Interface	`ZebOS.conf`; see Unified Configuration File	N/A	`2650/tcp`
`isisd`	Intermediate System-Intermediate System (IS-IS)	`isisd.conf`	`isisd.conf.sample`	`2609/tcp`
`l2mribd`	Layer 2 Multicast Routing Information Base	`l2mribd.conf`	N/A	2664/tcp
`lagd`	Link Aggregation Control Protocol (LACP) and Distributed Resilient Network Interconnect (DRNI)	`lagd.conf`	`lagd.conf.sample`	`2619/tcp`
`ldpd`	Label Distribution Protocol (LDP)	`ldpd.conf`	`ldpd.conf.sample`	`2607/tcp`
`lsmd`	Logical Storage Manager (LSM): manages logical switch routers (LSRs) in a multi-tenant environment	`ZebOS.conf`, `ZebOS_MT.conf`	N/A	`N/A`
`mribd`	Multicast Routing Information Base	`mribd.conf`	N/A	`2610/tcp`
`mstpd`	Multiple Spanning Tree Protocol (MSTP), Rapid Spanning Tree Protocol, and Spanning Tree Protocol	`mstpd.conf`	`mstp.conf.sample`	`2618/tcp`

Table 1: ZebOS-XP daemons (Continued)

Daemon name	Manages	Default configuration file	Sample file	Port Number
ndd	Neighbor discovery	ndd.conf	N/A	2661/tcp
nsm	Network Services Module	nsm.conf	nsm.conf.sample	2601/tcp
oamd	MPLS, Bidirectional Forwarding Detection (BFD), MPLS TP (ping; Li, Lb, LM, DM, FM and PS)	oamd.conf	N/A	N/A
onmd	Link Layer Discovery Protocol (LLDP), Ethernet to the First Mile (EFM), and Connectivity Fault Management (CFM)	onmd.conf	N/A	2651/tcp
ospf6d	OSPFv3 protocol	ospf6d.conf	ospf6d.conf.sample	2606/tcp
ospfd	Open Shortest Path First (OSPFv2) and Constrained Shortest Path First (CSPF) protocols	ospfd.conf	ospfd.conf.sample	2604/tcp
pimd	Protocol Independent Multicasting (PIM)	pimd.conf	pimd.conf.sample	2611/tcp
ptpd	Precision Time Protocol (PTP)	ptpd.conf	N/A	2655/tcp
ribd	Unicast Routing Information Base	ribd.conf	N/A	2662/tcp
ripd	Routing Information Protocol (RIPv1, RiPv2)	ripd.conf	ripd.conf.sample	2602/tcp
ripngd	Routing Information Protocol Next Generation (RIPng)	ripngd.conf	ripngd.conf.sample	2603/tcp
rmond	Remote Monitoring Management Information Base (MIB)	rmond.conf	N/A	2622/tcp
rsvpd	Resource ReSeservation Protocol (RSVP)	rsvpd.conf	rsvpd.conf.sample	2608/tcp
spbd	Shortest Path Bridging (SPB)	spbd.conf	N/A	2656/tcp
synced	Synchronous Ethernet	synced.conf	N/A	2657/tcp
trilld	TRansparent Interconnection of Lots of Links (TRILL)	trilld.conf	N/A	2654/tcp
vlogd	VR context-sensitive log-in	vlogd.conf	N/A	N/A
vrrpd	Virtual Router Redundancy Protocol (VRRP)	vrrpd.conf	vrrpd.conf.sample	2660/tcp

Starting ZebOS-XP Daemons

You can start the various ZebOS-XP daemons in any order. The general steps are listed below; details are in the sections that follow. If you stop NSM, you must restart all protocol daemons after you restart NSM.

The steps that follow describe how to start or stop ZebOS-XP daemons.

Note: To run a protocol module in the background, you must starts the daemon with the `-d` option only. ZebOS-XP does *not* support the "&" background operator.

1. Start ZebOS-XP daemons from the command prompt using the binary name and appropriate runtime flags. Use the setup that applies for your installation as described below.
   - If you did not enable the IMI (Integrated Management Interface) configuration option before compiling ZebOS-XP, enter these commands:

     ```
     ./nsm -d
     ./<protocol> -d
     ```
   - If you enabled the IMI configuration option, but did not enable the IMI Shell (IMISH) configuration option before compiling ZebOS-XP, enter these commands:

     ```
     ./nsm -d
     ./<protocol> -d
     ./imi -d
     ```
   - If you enabled the IMISH option (which also enables IMI) before compiling ZebOS-XP, enter these commands:

     ```
     ./nsm -d
     ./<protocol> -d
     ./imi -d
     ./imish
     ```

   Note: If you enabled the IMISH configuration before compiling ZebOS-XP, you cannot connect to the protocol daemons using the port number. Therefore, skip the telnet step and proceed to the following step.

2. Start a telnet session:

   ```
   telnet <IPADDRESS/localhost> <PORT>
   ```

   where

   `IPADDRESS` Is the address of the machine with the protocol daemons

   `<localhost>` Indicates the protocol daemons are running on the same machine

   `<PORT>` Is the port number of the protocol daemon

   Refer to Port Numbers for Daemons on page 47 for details.

   Note: When IMI is enabled, use port 2650 to access all protocols and NSM.

   Note: If ZebOS-XP is compiled without enabling IMISH, it is required to telnet to IMI by using the port number 2650 to get access to the device. Before starting the daemons, a file, `/usr/local/etc/ZebOS.conf`, must be created that includes the following command:

   ```
   password zebra
   ```

   After this step, start the daemons and connect to IMI by using the `telnet localhost 2650` command.

Stopping ZebOS-XP Daemons

Stop a ZebOS-XP daemon by terminating the process of the daemon, for example:

```
kill <pid>
```

where

`<pid>` is the Process ID of the daemon that is to be stopped.

Note: Use the `ps` command to find the process ID of a running daemon.

Configuration Files

Table 1 shows the sample configuration files provided by IP Infusion Inc.. You can edit the sample configuration files as required for your installation.

Location

By default, the sample configuration files are in the sub-folders of their respective module. However, when you start a daemon, you can load a configuration file from a different location as described in Loading Configuration Files.

Unified Configuration File

The `ZebOS.conf` file is a unified default configuration file used with the IMI daemon that includes the collective configuration of all protocol daemons.

Loading Configuration Files

You can use the `-f` option to load configuration files from a specific location. The criteria used for reading configuration files is described in the next sections.

Without -f Option

- When you start a daemon without the `-f` option, the configuration loads from the file in the working (current) directory.

  For example, if you start NSM:

  ```
  ./nsm -d
  ```

  the configuration loads from `nsm.conf` in the working directory.

- If a configuration file is not present in the working directory, the daemon loads the configuration from the `/usr/local/etc` directory.

  Note: When you start the IMI daemon without the `-f` option, IMI reads the configuration from the `ZebOS.conf` file in the working directory, or from `ZebOS.conf` in `/usr/local/etc`.

With -f Option

When you start a daemon with the `-f` option:

- If you specify the file with an absolute path, for example, `./nsm -d -f /etc/nsm.conf`, the configuration loads from the specified path and file.

- If you specify the file with a relative path, for example, `./nsm -d -f ../nsm.conf`, the file loads by concatenating its name with that of the current directory.
- If you specify the file without an absolute or relative path, for example, `./nsm -d -f nsm.conf`, the file loads by concatenating its name with that of the working directory.

Changing Configuration Files

Make configuration changes using ZebOS-XP commands as needed for each daemon, then save the changes using one of these commands:

```
write file
write memory
```

Use the `write file` command to save your configuration changes to the original configuration file.

For example, if you start the NSM daemon using the default configuration, the `write file` or `write memory` commands write the configuration to the `/usr/local/etc/nsm.conf` physical file.

Note: Configuration changes are lost if a daemon ends before you give either the `write file` or `write memory` command.

Creating or Editing Configuration Files

If you do not use the IMI shell (`imish`), you must create a configuration file for each of the ZebOS-XP daemons you need. IP Infusion Inc. supplies the sample configuration files listed in Table 1. You can use these sample files as guides. The minimum configuration commands required for each daemon are as follows:

`hostname <HOSTNAME>` The name of the daemon, for example, `ripd`
`password <PASSWORD>` The password required to run the daemon, for example, `zebos`

Note: The only time you need to edit a daemon configuration file is when you need to change the configuration. An example is provided below to guide you in this process.

To edit the `<name.conf>` daemon configuration file, perform the following steps.

1. Copy the daemon sample file to the daemon configuration file, for example:
   ```
   cp rip.conf.sample rip.conf
   ```
2. Edit the configuration file using the sample file as a guide:
   ```
   % vi rip.conf
   ```
3. Change any information necessary to conform to your network.
4. Save your changes.

Port Numbers for Daemons

You can connect to the daemons from a terminal emulator using the port numbers defined in `ZebOS/lib/vty.h` as shown in Table 1.

Appendix C Install and Build Debian 8

This appendix shows how to install and build Debian 8.

Prerequisites

- DVD ISO images for Debian GNU/Linux 8.2 (Jessie) for one of these processor architectures:
  - 64 bit (amd64): Required for one processor or SMP for 64 bit
  - 32 bit (i386): Required for one processor or SMP for 32 bit

  Download these from https://www.debian.org/CD/http-ftp.
- ZebOS-XP tar ball (`ZebOS.tar.gz`) delivered by IP Infusion Inc.

Installing the Base Operating System

Install the Debian 8 OS from the DVD ISO image on a machine as shown below.

- More than 50 GB hard disk when installing on a virtual machine
- Memory: 3 GB recommended
- Machine can have one of the following processor architectures. Select the Debian 8 Jessie DVD ISO image as mentioned in the prerequisites:
  - 32bit (i386) with one processor
  - 64bit (amd64) with one processoR
  - 32bit (i386) with more than one processors (SMP)
  - 64bit (amd64) with more than one processors (SMP)
- You can refer to https://www.debian.org/doc/ for details about Debian installation.

1. Edit `/etc/resolv.conf` and ensure that your nameserver (DNS) is specified and that the machine can connect to the Internet.
2. Edit `/etc/apt/sources.list` so that it contains *only* these two lines:

```
deb http://httpredir.debian.org/debian jessie main
deb-src http://httpredir.debian.org/debian jessie main
```

3. Install the following packages needed to compile:

```
apt-get update
apt-get install openssh-server
apt-get install unzip fakeroot uuid-runtime build-essential crash kexec-tools
makedumpfile kernel-wedge git-core libncurses5 libncurses5-dev libelf-dev
apt-get install asciidoc binutils-dev gcc-multilib libc6-dev-i386 swig equivs doxygen
bc devscripts openssl-* vim
apt-get update
```

4. Install the Linux kernel source package.

   For x86 (also for SMP):

   ```
   aptitude install linux-source-3.16=3.16.7-ckt11-1+deb8u3
   ```

   For amd64 (also for SMP):

   ```
   aptitude install linux-source-3.16=3.16.7-ckt11-1+deb8u5
   ```

Building the Kernel for Debian 8

1. Extract the kernel source in the `/usr/src` directory. You get the`linux-source-3.16.tar.xz` file when you install the Linux source package with the `aptitude install` command shown earlier.

   ```
   cd /usr/src
   tar -xf linux-source-3.16.tar.xz
   ```

2. Extract the ZebOS-XP tar ball.

   ```
   mkdir -p /root/build/
   cd /root/build
   tar -zxf ZebOS.tar.gz
   cd /usr/src/linux-source-3.16
   ```

3. Copy one of the configuration files for the respective processor architecture.

   For x86:

```
cp -rpf /root/build/ZebOS/kernel/linux/ZebOS_XP-3.16.7/configs/config-i386-3.16 .config
```

   For amd64 (64 bit):

```
cp -rpf /root/build/ZebOS/kernel/linux/ZebOS_XP-3.16.7/configs/config-x86_64-3.16
.config
```

   For x86 SMP:

```
cp -rpf /root/build/ZebOS/kernel/linux/ZebOS_XP-3.16.7/configs/config-i386-smp-3.16
.config
```

   For amd64 SMP:

```
cp -rpf /root/build/ZebOS/kernel/linux/ZebOS_XP-3.16.7/configs/config-x86_64-smp-3.16
.config
```

4. Apply the ZebOS-XP kernel patches.

   ```
   cd /usr/src/linux-source-3.16
   cat /root/build/ZebOS/kernel/linux/ZebOS_XP-3.16.7/patches/0*.patch >> all.patch
   patch -Np1 < ./all.patch
   ```

5. Create the Debian kernel packages.

   ```
   make deb-pkg LOCALVERSION=-xp-1.4.0.0 KDEB_PKGVERSION=0
   ```

   Once kernel compilation completes, the kernel Debian packages are created in the `/usr/src` directory with following format.

   ```
   linux-headers-*.deb
   linux-libc-dev_*.deb
   linux-image-*.deb
   ```

6. Install the kernel Debian packages.

   ```
   dpkg –i <file.deb>
   ```


`linux-headers-*.deb` must be installed to build ZebOS-XP

`linux-image-*.deb` must be installed to run ZebOS-XP

Appendix D Configure SNMP

ZebOS-XP supports the SMUX (SNMP Multiplexing) RFC 1227 and AgentX (Agent Extension) RFC 2741 protocols, which are used by SNMP (Simple Network Management Protocol) agents to query variables maintained by another user-level process. This chapter provides instructions for configuring SMUX and AgentX settings for use by SNMP. It also describes various SNMP utilities.

Note: You cannot use SMUX and AgentX protocols simultaneously. Follow the instructions provided in this chapter for the protocol you are using.

Obtain and Install SNMP

If you do not already have SNMP installed, follow these steps.

1. Download the SNMP software from `www.net-snmp.org` Web site.
2. Install and configure the software by following the instructions in the `INSTALL` file that comes with the software.
3. Proceed to one of the following sections to configure SNMP with either the SMUX or the AgentX protocol.

   Note: You cannot configure SNMP with *both* SMUX and AgentX. One or the other must be used.

Configure SNMP with SMUX

Before compiling SNMP, check the `SMUX_MAX_PEERS` macro in the file `smux.h`. This macro sets the value for the number of unique peers allowed in SNMP. Each change in the SMUX peer OID is counted as a peer. IP Infusion Inc. recommends a minimum value of 25. However, use a larger value if you require more frequent changes in SMUX OID,

To configure SNMP with the SMUX protocol:

1. Change to the directory containing the SNMP source code:

   ```
   cd snmp
   ```

2. Type the following command to configure SNMP with SMUX enabled for your system:

   ```
   ./configure --with-mib-modules="smux" --enable-shared
   ```

   Note: Press the `enter` key at all prompts during `snmpd` configuration.

3. To compile the software for your system, issue the `make` command in the root source directory:

   ```
   make
   umask 022
   ```

4. Install the build components by entering the following commands:

   ```
   make install
   ldconfig
   ```

   Note: Be sure to start the SNMP daemon (`snmpd`) after starting the `nsm` daemon, but *before* starting any protocol daemon that needs to communicate with SNMP.

Configure SMUX

Note: When ZebOS-XP is started, each enabled protocol daemon tries to connect to the SNMP daemon using the default SMUX peer ID associated with it. Thus, when using SMUX, the SNMP daemon must be configured with the default SMUX peer ID only.

1. Create a file `snmpd.conf` in the `/usr/local/share/snmp/` (default) directory.
2. When the SNMP daemon starts, it loads the configuration file from the default location. It can also load the configuration file from another directory. To load the `snmpd.conf` file from a different location, use the following command

   ```
   snmpd -d -c <PATH>/snmpd.conf -C
   ```

   where `<PATH>` is the directory from where the `snmpd.conf` file is to be loaded.
3. Add the following lines to the end of the file:

`smuxpeer 1.3.6.1.4.1.3317.1.2.3`	For RIP
`smuxpeer 1.3.6.1.4.1.3317.1.2.5`	For OSPF
`smuxpeer 1.3.6.1.4.1.3317.1.2.2`	For BGP
`smuxpeer 1.3.6.1.4.1.3317.1.2.8`	For PIM-SM
`smuxpeer 1.3.6.1.4.1.3317.1.2.6`	For LDP
`smuxpeer 1.3.6.1.4.1.3317.1.2.10`	For NSM
`smuxpeer 1.3.6.1.4.1.3317.1.2.11`	For OSPFv3
`smuxpeer 1.3.6.1.4.1.3317.1.2.7`	For ISISv4/v6
`smuxpeer 1.3.6.1.4.1.3317.1.2.9`	For RSVP-TE
`smuxpeer` 1.3.6.1.4.1.3317.1.2.15	For Authd
`smuxpeer` 1.2.840.10006.300.43	For LACP
`smuxpeer` 1.3.6.1.2.1.16	For RMON
`smuxpeer` 1.3.6.1.4.1.3317.1.2.16	For RSTP
`smuxpeer` 1.3.6.1.2.1.17	For Bridge
`smuxpeer` 1.3.6.1.4.1.3317.1.2.12	For STP
`smuxpeer` 1.3.6.1.4.1.3317.1.2.9	For mribd
`rwcommunity test`	

The `smuxpeer` communicates between the `snmpd` and the routing protocols. All `smuxpeer` values listed above are `smuxpeer IDs` assigned by IANA (Internet Assigned Numbers Authority). These values correspond with the ZebOS-XP routing protocols and should not be modified.

The string `rwcommunity test` is the community sitting between the `snmpd` (agent) and the physical SNMP management station (client). The string `test` is the community authentication key used for authorization between the agent and the client. Once your configuration is complete, this string should be changed for security purposes.

Enable SNMP in the Source Code

1. Install the router software on the machine and untar the source `.tar` file.
2. Change to the ZebOS-XP directory (the SNMP option is enabled by default) and run configure:

   ```
   ./configure
   ```

3. Change directory to `platform/PLATFORM`, where `PLATFORM` is the OS being used, for example:

   ```
   cd platform/LINUX
   ```

4. Compile the code by entering these (this) command(s):

   ```
   make all
   make install
   ```

Configure SNMP with AgentX

Note: Do not complete the steps in this section if you have configured SNMP with SMUX.

To configure SNMP with the AgentX protocol:

1. Change to the directory containing the SNMP source code.

2. Type the following command to configure `AgentX` for your system:

   ```
   ./configure --with-mib-modules="agentx"
   ```

   Note: The `configure` command configures the agent to use the `AgentX` protocol. This is an IETF-defined protocol that allows a master/client relationship between agents and subagents.

3. Compile the software for your system by typing the `make` command in the source root directory:

   ```
   make
   umask 022
   ```

4. Install the build components by typing the following command:

   ```
   make install
   ```

Configure AgentX

When ZebOS-XP is started, each daemon connects to the Master Agent running at the SNMP daemon by sending requests to the master agent. When a request is accepted by the master agent, a session is established between the subagent at the ZebOS-XP protocol daemon and the master agent running at the SNMP daemon. These sessions are used by the protocol daemons to communicate with the SNMP daemon.

1. Create a file called `snmpd.conf` in the `/usr/local/share/snmp/` directory (default location). When the SNMP daemon starts, it loads the configuration file from the default location.

2. SNMP can also load the configuration file from another directory. To load the `snmpd.conf` file from a different location, use the following command:

   ```
   snmpd -x localhost:705 -c <PATH>/snmpd.conf -C
   ```

   where `<PATH>` is the directory from where the `snmpd.conf` file is located.

3. Add this line to the end of the file:

   ```
   rwcommunity test
   ```

   This is the community setting between the `snmpd` (agent) and the actual SNMP Management Station, or client. The string `test` is an authentication key used for authorizing communication between the agent and the client. Once you have finished configuring AgentX, you should change this password for security purposes.

4. Add another line to the end of the file:

   ```
   master agentx
   ```

   This tells the agents to behave as the master in the master/client `AgentX` protocol.

Enable AgentX in the Source Code

1. Install the router software on the machine and unzip the source-code tar file.
2. Change to the ZebOS-XP directory, run configure with `Agentx` enabled (apart from the other options) and compile the code:

   ```
   ./configure --enable-agentx
   ```

3. Change directory to `platform/PLATFORM,` where `PLATFORM` is the OS being used, for example:

   ```
   cd platform/LINUX
   ```

4. Compile the code by entering these (this) command(s):

   ```
   make all
   make install
   ```

   Note: If you do not enable AgentX, the SNMP code is built with SMUX enabled by default.

SNMP Utilities

This section applies to both the SMUX and AgentX protocols. In the following sections:

`-c` is an option specifying community string

`test` is the community string authentication key

`10.10.10.50` is the agent machine's IP address, which can be a localhost

`i =` is used to set an integer value

`a =` is used to set an IP address

`s =` is used to set a string value

`TYPE = i|a|s`

`VALUE =` Value of the specified Type. For example, `i 20` sets an integer value of 20.

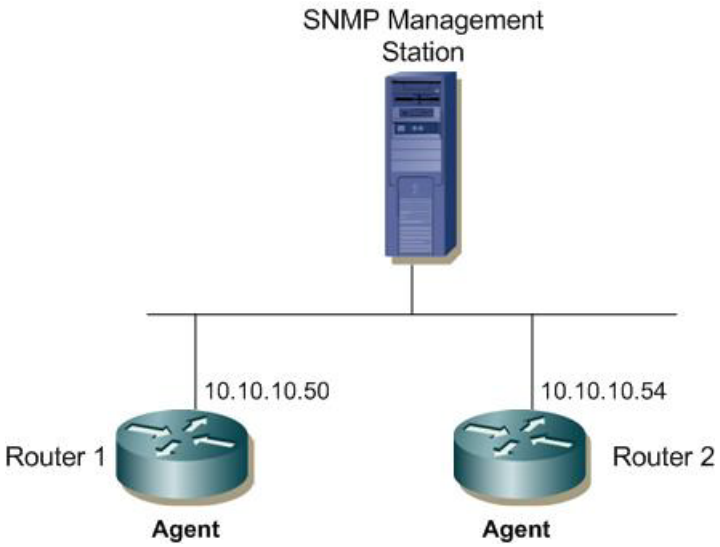

Figure D-1: SNMP utilities

OSPF

snmpget

From the client or management, execute `snmpget` to get the value from the OSPF MIB.

```
snmpget -c test 10.10.10.50 .1.3.6.1.2.1.14.1.1.0
```

`.1.3.6.1.2.1.14.1.1.0` is the object ID of the variable `ospfRouterId`

snmpgetnext

From the client or management, execute `snmpgetnext` to get the next value from the OSPF MIB.

```
snmpgetnext -c test 10.10.10.50 .1.3.6.1.2.1.14.1.1.0
```

Executing this command returns `ospfAdminStat .1.3.6.1.2.1.14.1.2.0`, the next Object ID from `ospfGeneralGroup` table.

snmpset

From the management station execute `snmpset` to set the router ID value to 1.1.1.1. This command is also used to create and delete the variable in MIB.

```
snmpset -c test 10.10.10.50 .1.3.6.1.2.1.14.1.1.0 a 1.1.1.1
```

snmpwalk

Execute `snmpwalk` to walk through each table in the OSPF MIB.

```
snmpwalk -c test 10.10.10.50 .1.3.6.1.2.1.14.7.1.6.10.10.10.50.3
```

`.1.3.6.1.2.1.14.7.1.6.10.10.10.50.3` is the Object ID of the variable `ospfIfRtrPriority`.

OSPFv3

snmpget

From the client or management, execute `snmpget` to get the value from the OSPFv3 MIB.

```
snmpget -c test 10.10.10.50 .1.3.6.1.3.122.1.1.2.0
```

`.1.3.6.1.3.122.1.1.2.0` is the Object ID of the variable `ospfAdminStat`.

snmpgetnext

From the client or management, execute `snmpgetnext` to get the next value from the OSPFv3 MIB.

```
snmpgetnext -c test 10.10.10.50 .1.3.6.1.3.122.1.1.1.0
```

Executing this command returns `ospfAdminStat .1.3.6.1.3.122.1.1.2.0`, the next Object ID from `ospfGeneralGroup` table.

snmpwalk

Execute `snmpwalk` to walk through each table in the OSPFv3 MIB.

```
snmpwalk -c test 10.10.10.50 .1.3.6.1.3.122.1.1.3.0
```

`.1.3.6.1.3.122.1.1.3.0` is the Object ID for the variable `ospfVersionNumber`.

IS-IS

snmpget

The following command gets the value of `isissysinstance` from the `ISISSysTable`.

```
snmpget -c test 10.10.10.50 .1.3.6.1.3.37.1.1.1.1.1.1
```

snmpwalk

The following command walks through the `ISISSysTable`.

```
snmpwalk -c test 10.10.10.50 .1.3.6.1.3.37.1.1
```

snmpset

From the management station execute `snmpset` to set `isisSysType` to be Level-2-only.

```
snmpset -c test 10.10.10.68 .1.3.6.1.3.37.1.1.1.1.3.1 i 2
```

BGP

snmpget

The following command gets the value of `bgpVersion`.

```
snmpget -c test 10.10.10.50 .1.3.6.1.2.1.15.1.0
```

snmpset

The following command sets the value of `bgpPeerMinRouteAdvertisementInterval`.

```
snmpset -c test 10.10.10.50 .1.3.6.1.2.1.15.3.1.23.10.10.10.52 i 40
```

snmpwalk

The following command walks through `bgpPeerTable`.

```
snmpwalk -c test 10.10.10.50 .1.3.6.1.2.1.15.1
```

RIP

snmpget

The following command gets the value of `rip2GlobalRouteChanges`.

```
snmpget -c test 10.10.10.50 .1.3.6.1.2.1.23.1.1.0
```

snmpwalk

The following command walks through `ripInterfaceTable`.

```
snmpwalk -c test 10.10.10.50 .1.3.6.1.2.1.23.1
```

snmpset

The following command sets the value of `rip2IfConfSend`

```
snmpset -c test 10.10.10.50 .1.3.6.1.2.1.23.3.1.5.10.10.11.50 i 1
```

PIM-SM

snmpget

The following command gets the value of `pimJoinPruneInterval` which is the default interval at which PIM-SM Join/Prune messages are to be sent.

```
snmpget -c test 10.10.10.50 .1.3.6.1.3.61.1.1.1.0
```

snmpset

The following command sets the value of `pimJoinPruneInterval` which is the default interval at which PIM-SM Join/Prune messages are to be sent.

```
snmpset -c test 10.10.10.50 .1.3.6.1.3.61.1.1.1.0 i 30
```

snmpwalk

The following command walks through the PIM Interface table

```
snmpwalk -c test 10.10.10.50 1.3.6.1.3.61.1.1.2
```

LDP

snmpget

`snmpget` on LDP returns `mplsLdpLsrId` from `mplsLdpLsrObjects:`

```
snmpget -c test 10.10.10.10 .1.3.6.1.2.1.10.1.4.1.1.1.0
```

Output:

```
transmission.1.4.1.1.1.0 =  Hex: C0 A8 00 2A
```

snmpwalk

`snmpwalk` on LDP returns `mplsLdpEntityGenericObjects` from `mplsLdpEntityObjects:`

```
snmpwalk -c test 10.10.10.10 .1.3.6.1.2.1.10.1.4.1.2.5
```

Sample output

```
transmission.1.4.1.2.5.1.1.3.192.168.0.42.0.0.1.16.1048575 = 0
transmission.1.4.1.2.5.1.1.4.192.168.0.42.0.0.1.16.1048575 = 2
transmission.1.4.1.2.5.1.1.5.192.168.0.42.0.0.1.16.1048575 = 1
```

snmpgetnext

`snmpgetnext` on LDP returns `mplsLdpLsrLoopDetectionCapable` from `mplsLdpLsrObjects:`

```
snmpgetnext -c test 10.10.10.10 .1.3.6.1.2.1.10.1.4.1.1.1.0
```

Sample output

```
transmission.1.4.1.1.2.0 = 5
```

NSM (LSR)

snmpget

`snmpget` on LSR MIB returns `mplsInterfaceLabelMaxOut` from `mplsInterfaceConfTable`.

```
snmpget -c test 10.10.10.100 .1.3.6.1.2.1.10.1.2.1.1.1.3.1
```

Sample output

```
transmission.1.2.1.1.1.3.1 = 1048575
```

snmpwalk

`snmpwalk` on LSR MIB returns the `mplsInterfaceConfTable`:

```
snmpwalk -c test 10.10.10.10 .1.3.6.1.2.1.10.1.2.1.1
```

Sample Output

```
transmission.1.2.1.1.1.2.0 = 16
transmission.1.2.1.1.1.2.2 = 16
transmission.1.2.1.1.1.2.3 = 16
transmission.1.2.1.1.1.2.4 = 16
transmission.1.2.1.1.1.3.0 = 1048575
transmission.1.2.1.1.1.3.2 = 1048575
transmission.1.2.1.1.1.3.3 = 1048575
transmission.1.2.1.1.1.3.4 = 1048575
transmission.1.2.1.1.1.4.0 = 16
transmission.1.2.1.1.1.4.2 = 16
transmission.1.2.1.1.1.4.3 = 16
transmission.1.2.1.1.1.4.4 = 16
transmission.1.2.1.1.1.5.0 = 1048575
transmission.1.2.1.1.1.5.2 = 1048575
transmission.1.2.1.1.1.5.3 = 1048575
transmission.1.2.1.1.1.5.4 = 1048575
transmission.1.2.1.1.1.6.2 = 0
transmission.1.2.1.1.1.6.3 = 100000
transmission.1.2.1.1.1.6.4 = 100000
transmission.1.2.1.1.1.7.2 = 0
transmission.1.2.1.1.1.7.3 = 100000
transmission.1.2.1.1.1.7.4 = 100000
transmission.1.2.1.1.1.8.0 = 1
transmission.1.2.1.1.1.8.2 = 1
transmission.1.2.1.1.1.8.3 = 1
transmission.1.2.1.1.1.8.4 = 1
```

snmpgetnext

`snmpgetnext` on LSR MIB returns the next value from `mplsInterfaceConfTable`:

```
snmpgetnext -c test 10.10.10.100 .1.3.6.1.2.1.10.1.2.1.1.1.3.1
```

Sample output

```
transmission.1.2.1.1.1.3.2 = 1048575
```

NSM (FTN)

snmpwalk

`snmpwalk` on FTN MIB returns the entire FTN MIB:

```
snmpwalk -c test 10.10.10.10 .1.3.6.1.2.1.10.1.5
```

Sample output

```
transmission.1.5.1.2.1.2.1 = 1
transmission.1.5.1.2.1.3.1 = ""
transmission.1.5.1.2.1.4.1 = 1
transmission.1.5.1.2.1.5.1 = 1
transmission.1.5.1.2.1.6.1 = ""
transmission.1.5.1.2.1.7.1 = ""
transmission.1.5.1.2.1.8.1 = "1.1.1.1"
transmission.1.5.1.2.1.9.1 = "1.1.1.1"
transmission.1.5.1.2.1.10.1 = 0
transmission.1.5.1.2.1.11.1 = 0
transmission.1.5.1.2.1.12.1 = 0
transmission.1.5.1.2.1.13.1 = 0
transmission.1.5.1.2.1.14.1 = 0
transmission.1.5.1.2.1.15.1 = 2
transmission.1.5.1.2.1.16.1 = "mplsFTNTable.1.0.0.1"
transmission.1.5.1.2.1.17.1 = 0
transmission.1.5.1.2.1.18.1 = 2
transmission.1.5.1.4.1.1.1 = 0
transmission.1.5.1.4.1.2.1 = 0
transmission.1.5.1.4.1.3.1 = 0
transmission.1.5.1.4.1.4.1 = 0
```

snmpget

`snmpget` on FTN MIB returns `mplsFTNDestipv4AddrMin` from `mplsFTNEntry:`

```
snmpget -c test 10.10.10.100 .1.3.6.1.2.1.10.1.5.1.2.1.8.1
```

Sample output:

```
transmission.1.5.1.2.1.8.1 = "1.1.1.1"
```

snmpgetnext

`snmpgetnext` on FTN MIB returns `mplsFTNDestipv4AddrMax` from `mplsFTNEntry:`

```
snmpgetnext -c test 10.10.10.10 .1.3.6.1.2.1.10.1.5.1.2.1.8.1
```

Sample output

```
transmission.1.5.1.2.1.9.1 = “1.1.1.1”
```

NSM (PW-MIB)

Pre-requisite Setup

1. Pre-configure the underlying PSN tunnel and switchport.

2. In pwTable (5601), configure the following MIB objects:
   - pwOwner
   - pwType
   - pwPeerAddr
   - pwId
   - pwName
   - pwOutboundLabel (required if pwOwner is manual)
   - pwInboundLabel (required if pwOwner is manual)
3. Make pwRowStatus active through SNMP.
4. When pwRowStatus (5601) goes active, a row is created in 5602 PSN-specific tables and in pwEnetTable (5603) with default values.
5. If the underlying PSN tunnel is RSVP, configure pwMplsMplsType as mplsTe in pwMplsTable and set a value for pwMplsOutboudTunnelIndex in pwMplsOutboundTable in 5602 through SNMP. Otherwise, you do not need to configure anything in 5602.
6. In pwEnetTable (5603) configure the following MIB objects:
   - pwEnetPwVlan
   - pwEnetVlanMode
   - pwEnetPortVlan
   - pwEnetPortIfIndex
7. Make pwEnetRowStatus active through SNMP.
8. Do a SNMP walk on 5601, 5602, and 5603 as shown below.

snmpwalk for 5601 MIB

snmpwalk on pwTable OR 5601 MIB returns all values in pwTable

```
snmpwalk -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.10.246.1.2.1
```

Sample Output

```
iso.3.6.1.2.1.10.246.1.2.1.2.1 = INTEGER: 5
iso.3.6.1.2.1.10.246.1.2.1.3.1 = INTEGER: 1
iso.3.6.1.2.1.10.246.1.2.1.4.1 = INTEGER: 1
iso.3.6.1.2.1.10.246.1.2.1.5.1 = INTEGER: 0
iso.3.6.1.2.1.10.246.1.2.1.6.1 = INTEGER: 0
iso.3.6.1.2.1.10.246.1.2.1.8.1 = INTEGER: 1
iso.3.6.1.2.1.10.246.1.2.1.9.1 = STRING: "1.1.1.1"
iso.3.6.1.2.1.10.246.1.2.1.10.1 = INTEGER: 0
iso.3.6.1.2.1.10.246.1.2.1.11.1 = INTEGER: 0
iso.3.6.1.2.1.10.246.1.2.1.12.1 = Gauge32: 101
iso.3.6.1.2.1.10.246.1.2.1.13.1 = Gauge32: 0
iso.3.6.1.2.1.10.246.1.2.1.14.1 = STRING: "0"
iso.3.6.1.2.1.10.246.1.2.1.15.1 = STRING: "0"
iso.3.6.1.2.1.10.246.1.2.1.16.1 = STRING: "0"
iso.3.6.1.2.1.10.246.1.2.1.17.1 = INTEGER: 2
iso.3.6.1.2.1.10.246.1.2.1.18.1 = INTEGER: 1500
```

```
iso.3.6.1.2.1.10.246.1.2.1.19.1 = INTEGER: 2
iso.3.6.1.2.1.10.246.1.2.1.20.1 = Hex-STRING: 00
iso.3.6.1.2.1.10.246.1.2.1.21.1 = Gauge32: 0
iso.3.6.1.2.1.10.246.1.2.1.22.1 = INTEGER: 6
iso.3.6.1.2.1.10.246.1.2.1.23.1 = INTEGER: 0
iso.3.6.1.2.1.10.246.1.2.1.24.1 = Hex-STRING: 00 00 00 00
iso.3.6.1.2.1.10.246.1.2.1.25.1 = Hex-STRING: 00
iso.3.6.1.2.1.10.246.1.2.1.26.1 = INTEGER: 0
iso.3.6.1.2.1.10.246.1.2.1.27.1 = Hex-STRING: 80
iso.3.6.1.2.1.10.246.1.2.1.28.1 = INTEGER: 1
iso.3.6.1.2.1.10.246.1.2.1.29.1 = Hex-STRING: 10
iso.3.6.1.2.1.10.246.1.2.1.30.1 = INTEGER: 1111
iso.3.6.1.2.1.10.246.1.2.1.31.1 = INTEGER: 2222
iso.3.6.1.2.1.10.246.1.2.1.32.1 = STRING: "VC-1"
iso.3.6.1.2.1.10.246.1.2.1.33.1 = STRING: "PE-2 to PE-1"
iso.3.6.1.2.1.10.246.1.2.1.34.1 = Timeticks: (700) 0:00:07.00
iso.3.6.1.2.1.10.246.1.2.1.35.1 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.10.246.1.2.1.36.1 = Timeticks: (700) 0:00:07.00
iso.3.6.1.2.1.10.246.1.2.1.37.1 = INTEGER: 0
iso.3.6.1.2.1.10.246.1.2.1.38.1 = INTEGER: 2
iso.3.6.1.2.1.10.246.1.2.1.39.1 = Hex-STRING: 80
iso.3.6.1.2.1.10.246.1.2.1.40.1 = INTEGER: 1
iso.3.6.1.2.1.10.246.1.2.1.41.1 = Hex-STRING: 80
iso.3.6.1.2.1.10.246.1.2.1.42.1 = INTEGER: 7
iso.3.6.1.2.1.10.246.1.2.1.43.1 = INTEGER: 1
iso.3.6.1.2.1.10.246.1.2.1.44.1 = INTEGER: 1
iso.3.6.1.2.1.10.246.1.2.1.45.1 = INTEGER: 2
iso.3.6.1.2.1.10.246.1.2.1.46.1 = INTEGER: 1
```

snmpget for 5601

`snmpget` on 5601 MIB returns `pwName` from `pwTable`.

```
snmpget -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.10.246.1.2.1.32.1
```

Sample Output

```
iso.3.6.1.2.1.10.246.1.2.1.32.1 = STRING: "VC-1"
```

snmpgetnext for 5601

`snmpgetnext` on 5601 MIB returns `pwDescr` from `pwTable`.

```
snmpgetnext -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.10.246.1.2.1.32.1
```

Sample Output

```
iso.3.6.1.2.1.10.246.1.2.1.33.1 = STRING: "PE-2 to PE-1"
```

snmpwalk for 5602 MIB

`snmpwalk` on `pwMplsTable` OR 5602 MIB returns all values in `pwMplsTable`.

```
snmpwalk -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.181.1.1.1
```

Sample Output

```
iso.3.6.1.2.1.181.1.1.1.1.1 = Hex-STRING: 80
```

```
iso.3.6.1.2.1.181.1.1.1.2.1 = INTEGER: 1
iso.3.6.1.2.1.181.1.1.1.3.1 = INTEGER: 0
iso.3.6.1.2.1.181.1.1.1.4.1 = INTEGER: 2
iso.3.6.1.2.1.181.1.1.1.5.1 = STRING: "0.0.0.0"
iso.3.6.1.2.1.181.1.1.1.6.1 = INTEGER: 0
iso.3.6.1.2.1.181.1.1.1.7.1 = STRING: "0.0.0.0"
iso.3.6.1.2.1.181.1.1.1.8.1 = INTEGER: 2
```

snmpget for 5602

`snmpget` on 5602 MIB returns `pwMplsMplsType` from `pwMplsTable`.

```
snmpget -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.181.1.1.1.1.1
```

Sample Output

```
iso.3.6.1.2.1.181.1.1.1.1.1 = Hex-STRING: 80
```

snmpgetnext for 5602

`snmpgetnext` on 5602 MIB returns `pwMplsExpBitsMode` from `pwMplsTable`.

```
snmpgetnext -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.181.1.1.1.1.1
```

Sample Output

```
iso.3.6.1.2.1.181.1.1.1.2.1 = INTEGER: 1
```

snmpwalk for 5602 MIB

`snmpwalk` on `pwMplsOutboundTable` OR 5602 MIB returns all values in `pwMplsOutboundTable`.

```
snmpwalk -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.181.1.2.1
```

Sample Output

```
iso.3.6.1.2.1.181.1.2.1.1.1 = INTEGER: 0
iso.3.6.1.2.1.181.1.2.1.2.1 = INTEGER: 101
iso.3.6.1.2.1.181.1.2.1.3.1 = INTEGER: 0
iso.3.6.1.2.1.181.1.2.1.4.1 = STRING: "0.0.0.0"
iso.3.6.1.2.1.181.1.2.1.5.1 = STRING: "1.1.1.1"
iso.3.6.1.2.1.181.1.2.1.6.1 = INTEGER: 0
iso.3.6.1.2.1.181.1.2.1.7.1 = INTEGER: 2
```

snmpget for 5602

`snmpget` on 5602 MIB returns `pwMplsOutboundLcrXcIndex` from `pwMplsOutboundTable`.

```
snmpget -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.181.1.2.1.1.1
```

Sample Output

```
iso.3.6.1.2.1.181.1.2.1.1.1 = INTEGER: 0
```

snmpgetnext for 5602

`snmpgetnext` on 5602 MIB returns `pwMplsOutboundTunnelIndex` from `pwMplsOutboundTable`.

```
snmpgetnext -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.181.1.2.1.1.1
```

Sample Output

```
iso.3.6.1.2.1.181.1.2.1.2.1 = INTEGER: 0
```

snmpwalk for 5603

`snmpwalk` on `pwEnetTable` OR 5603 MIB returns all values in `pwEnetTable`.

```
snmpwalk -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.180.1.1.1
```

Sample Output

```
iso.3.6.1.2.1.180.1.1.1.2.1.1 = INTEGER: 4095
iso.3.6.1.2.1.180.1.1.1.3.1.1 = INTEGER: 1
iso.3.6.1.2.1.180.1.1.1.4.1.1 = INTEGER: 4095
iso.3.6.1.2.1.180.1.1.1.5.1.1 = INTEGER: 3
iso.3.6.1.2.1.180.1.1.1.6.1.1 = INTEGER: 0
iso.3.6.1.2.1.180.1.1.1.7.1.1 = INTEGER: 1
iso.3.6.1.2.1.180.1.1.1.8.1.1 = INTEGER: 2
```

snmpget for 5603

`snmpget` on 5603 MIB returns `pwEnetPwVlan` from `pwEnetTable`.

```
snmpget -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.180.1.1.1.2.1.1
```

Sample Output

```
iso.3.6.1.2.1.180.1.1.1.2.1.1 = INTEGER: 4095
```

snmpgetnext for 5603

`snmpgetnext` on 5603 MIB returns `pwEnetVlanMode` from `pwEnetTable`.

```
snmpgetnext -v 2c -c test 10.12.17.36 .1.3.6.1.2.1.180.1.1.1.2.1.1
```

Sample Output

```
iso.3.6.1.2.1.180.1.1.1.3.1.1 = INTEGER: 1
```

Configuring LSR and FTN through SNMP

RFC3813 supports following tables:

- MplsInterfaceTable
- MplsInsegmentTable
- MplsOutSegmentTable
- MplsXCtable

RFC3814 supports following tables:

- MplsFTNTable- Configure the table and make row status active through snmp.

Pre-requisite to be completed beofre configuring LSR and FTN

1. Pre-configure routing protocol OR static routes.
2. Pre-enable basic configuration, label switching on interfaces used for ILM and FTN entry.
3. Pre-configure each table with appropriate value and change the *rowstatus* status to active through SNMP.

SNMPSET to create an Insegment Table

Configure the below objects using MIBs and SNMPset in `mplsinsegment` table.

- `mplsinsegmentinterface`
- `mplsinsegmentlabel`
- `mplsoutsegmentrowstatus`

```
[root@localhost sbin]# snmpset -v2c -c test 10.12.48.52
  1.3.6.1.2.1.10.166.2.1.4.1.10.1 i  5
iso.3.6.1.2.1.10.166.2.1.4.1.10.1 = INTEGER: 5
```

To make active

```
[root@localhost sbin]# snmpset -v2c -c test 10.12.48.52
  1.3.6.1.2.1.10.166.2.1.4.1.10.1 i  1
iso.3.6.1.2.1.10.166.2.1.4.1.10.1 = INTEGER: 1
```

SNMP WALK for mplsinsegmentTable

`snmpwalk` on `mplsInsegmentTable` will returns entire values in `mplsInsegmentTable`

```
snmpwalk -v2c -c test 10.12.48.52 1.3.6.1.2.1.10.166.2.1.4.1
```

Sample output

```
[root@localhost sbin]# snmpwalk -v2c -c test 10.12.48.52 1.3.6.1.2.1.10.166.2.1.4.1
  iso.3.6.1.2.1.10.166.2.1.4.1.2.1 = INTEGER: 3
  iso.3.6.1.2.1.10.166.2.1.4.1.3.1 = Gauge32: 100
  iso.3.6.1.2.1.10.166.2.1.4.1.4.1 = STRING: "mplsLbPtr.0.0"
  iso.3.6.1.2.1.10.166.2.1.4.1.5.1 = INTEGER: 1
  iso.3.6.1.2.1.10.166.2.1.4.1.6.1 = INTEGER: 1
  iso.3.6.1.2.1.10.166.2.1.4.1.7.1 = INTEGER: 0
  iso.3.6.1.2.1.10.166.2.1.4.1.8.1 = INTEGER: 3
  iso.3.6.1.2.1.10.166.2.1.4.1.9.1 = STRING: "mplsTnRscPtr.0.0"
```

```
  iso.3.6.1.2.1.10.166.2.1.4.1.10.1 = INTEGER: 1
  iso.3.6.1.2.1.10.166.2.1.4.1.11.1 = INTEGER: 2
```

`snmpget` on `mplsInsegmentInterface` MIB object will return the `mplsInsegmentInterface` value from `mplsInsegmentlTable`

```
[root@localhost sbin]# snmpget -v2c -c test 10.12.48.52
  1.3.6.1.2.1.10.166.2.1.4.1.2.1
iso.3.6.1.2.1.10.166.2.1.4.1.2.1 = INTEGER: 3
```

`snmpgetnext` on `mplsInsegmentInterfaceMIB` object will return the `mplsInsegmentLabel` value from mplsInsegmentTable

```
[root@localhost sbin]# snmpgetnext -v2c -c test 10.12.48.52
  1.3.6.1.2.1.10.166.2.1.4.1.2.1
iso.3.6.1.2.1.10.166.2.1.4.1.3.1 = Gauge32: 100
```

SNMP WALK for mplOutSegmentTable

`mplsoutsegment` table configures the below MIBS using snmpset:

- mplsOutsegmentinterface
- mplsoutsegmenttoplabel
- mplsoutsegmentnexthopaddr
- mplsoutsegmentrowstatus

```
[root@localhost sbin]# snmpwalk -v2c -c test 10.12.48.52 1.3.6.1.2.1.10.166.2.1.7.1
  iso.3.6.1.2.1.10.166.2.1.7.1.2.1 = INTEGER: 4
  iso.3.6.1.2.1.10.166.2.1.7.1.3.1 = INTEGER: 1
  iso.3.6.1.2.1.10.166.2.1.7.1.4.1 = Gauge32: 200
  iso.3.6.1.2.1.10.166.2.1.7.1.5.1 = STRING: "mplsOutSegLbPtr.0.0"
  iso.3.6.1.2.1.10.166.2.1.7.1.6.1 = INTEGER: 1
  iso.3.6.1.2.1.10.166.2.1.7.1.7.1 = IpAddress: 2.2.2.2
  iso.3.6.1.2.1.10.166.2.1.7.1.8.1 = INTEGER: 0
  iso.3.6.1.2.1.10.166.2.1.7.1.9.1 = INTEGER: 3
  iso.3.6.1.2.1.10.166.2.1.7.1.10.1 = STRING: "mplsOutSegTnRscPtr.0.0"
  iso.3.6.1.2.1.10.166.2.1.7.1.11.1 = INTEGER: 1
  iso.3.6.1.2.1.10.166.2.1.7.1.12.1 = INTEGER: 2
```

SNMP WALK for mplsXCTable

Pre-requisites:

- Create XC table for ILM and FTN entry.

  For ex: for ILM entry use index as 1.1.1 and for FTN entry index should be 1.0.1 (Xcindex.insegindex.outsegindex)

- Create table and make rowstatus active.

```
[root@localhost sbin]# snmpwalk -v2c -c test 10.12.48.52
  1.3.6.1.2.1.10.166.2.1.10.1
  iso.3.6.1.2.1.10.166.2.1.10.1.4.1.0.1 = STRING: "0"
  iso.3.6.1.2.1.10.166.2.1.10.1.5.1.0.1 = INTEGER: 0
  iso.3.6.1.2.1.10.166.2.1.10.1.6.1.0.1 = INTEGER: 3
  iso.3.6.1.2.1.10.166.2.1.10.1.7.1.0.1 = INTEGER: 1
  iso.3.6.1.2.1.10.166.2.1.10.1.8.1.0.1 = INTEGER: 2
```

```
iso.3.6.1.2.1.10.166.2.1.10.1.9.1.0.1 = INTEGER: 1
iso.3.6.1.2.1.10.166.2.1.10.1.10.1.0.1 = INTEGER: 2
```

SNMP WALK for mplsFTNTable

FTN table configures the below MIBS using snmpset:

- Mplsftndescr
- Mplsftndestaddrmin
- Mplsftndestaddrmax
- Mplsftnactionpointer
- Mplsftnrowstatus

```
[root@localhost sbin]# snmpwalk -v2c -c test 10.12.48.52 1.3.6.1.2.1.10.166.8.1.3
iso.3.6.1.2.1.10.166.8.1.3.1.2.1 = INTEGER: 1
iso.3.6.1.2.1.10.166.8.1.3.1.3.1 = STRING: "mplssnmp"
iso.3.6.1.2.1.10.166.8.1.3.1.4.1 = STRING: "@"
iso.3.6.1.2.1.10.166.8.1.3.1.5.1 = INTEGER: 1
iso.3.6.1.2.1.10.166.8.1.3.1.6.1 = STRING: "0.0.0.0"
iso.3.6.1.2.1.10.166.8.1.3.1.7.1 = STRING: "0.0.0.0"
iso.3.6.1.2.1.10.166.8.1.3.1.8.1 = STRING: "10.0.0.0"
iso.3.6.1.2.1.10.166.8.1.3.1.9.1 = STRING: "10.0.0.63"
iso.3.6.1.2.1.10.166.8.1.3.1.10.1 = INTEGER: 0
iso.3.6.1.2.1.10.166.8.1.3.1.11.1 = INTEGER: 65535
iso.3.6.1.2.1.10.166.8.1.3.1.12.1 = INTEGER: 0
iso.3.6.1.2.1.10.166.8.1.3.1.13.1 = INTEGER: 65535
iso.3.6.1.2.1.10.166.8.1.3.1.14.1 = INTEGER: 255
iso.3.6.1.2.1.10.166.8.1.3.1.15.1 = INTEGER: 0
iso.3.6.1.2.1.10.166.8.1.3.1.16.1 = INTEGER: 1
iso.3.6.1.2.1.10.166.8.1.3.1.17.1 = STRING: "mplsXCTable.1.0.1"
iso.3.6.1.2.1.10.166.8.1.3.1.18.1 = INTEGER: 2
```

`snmpget` on `mplsftndescr` MIB object will return the `mplsdescr` value from `mplsftnTable`

```
[root@localhost sbin]# snmpget -v2c -c test 10.12.48.52
  1.3.6.1.2.1.10.166.8.1.3.1.3.1
iso.3.6.1.2.1.10.166.8.1.3.1.3.1 = STRING: "mplssnmp"
```

`snmpgetnext` on `mplsftnactionrype` MIB object will return `mplsftnactionpointer` value from `mplsftnTable`

```
[root@localhost sbin]# snmpgetnext -v2c -c test 10.12.48.52
  1.3.6.1.2.1.10.166.8.1.3.1.16.1
iso.3.6.1.2.1.10.166.8.1.3.1.17.1 = STRING: "mplsXCTable.1.0.1"
```

Configuring RSVP-TE-MIB through SNMP

RFC3812 supports following tables

- mplsTunnelTable
- mplsTunnelResourceTable
- mplsTunnelHopTable
- mplsTunnelPerfTable

Pre-requisite:

1. Pre-configure routing protocol OR static routes
2. Pre-configure bandwidth class and basic RSVP configuration on router and interface

SNMP WALK for mplsTunnelTable

Pre-requisites:

1. Following MIB objects needs to be configured and then make mplsTunnelRowStatus as active through SNMP.
   - mplsTunnelName
   - mplsTunnelRole
   - mplsTunnelHopTableIndex → required if tunnel is traveling through explicit path
   - mplsTunnelResourcePointer → required if bandwidth needs for the configured tunnel

snmpwalk on `mplsTunnelTable` will returns entire values in `mplsTunnelTable`

```
snmpwalk -v 2c -c test 10.12.17.37 .1.3.6.1.2.1.10.166.3.2.2.1
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.2.1.5.101.101.10.12.17.37.3.3.3.3 = STRING: "Tnl-1"
iso.3.6.1.2.1.10.166.3.2.2.1.6.101.101.10.12.17.37.3.3.3.3 = STRING: "PE-1 to PE-2"
iso.3.6.1.2.1.10.166.3.2.2.1.7.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.8.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.9.101.101.10.12.17.37.3.3.3.3 = INTEGER: 4
iso.3.6.1.2.1.10.166.3.2.2.1.10.101.101.10.12.17.37.3.3.3.3 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.2.1.11.101.101.10.12.17.37.3.3.3.3 = STRING:
  "mplsXCTable.0.0.0.1"
iso.3.6.1.2.1.10.166.3.2.2.1.12.101.101.10.12.17.37.3.3.3.3 = INTEGER: 2
iso.3.6.1.2.1.10.166.3.2.2.1.13.101.101.10.12.17.37.3.3.3.3 = INTEGER: 7
iso.3.6.1.2.1.10.166.3.2.2.1.14.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.15.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.16.101.101.10.12.17.37.3.3.3.3 = INTEGER: 2
iso.3.6.1.2.1.10.166.3.2.2.1.17.101.101.10.12.17.37.3.3.3.3 = STRING:
  "mplsRSTable.0"
iso.3.6.1.2.1.10.166.3.2.2.1.18.101.101.10.12.17.37.3.3.3.3 = INTEGER: 101
iso.3.6.1.2.1.10.166.3.2.2.1.19.101.101.10.12.17.37.3.3.3.3 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.2.1.20.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
```

```
iso.3.6.1.2.1.10.166.3.2.2.1.21.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.22.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.23.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.24.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.25.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.26.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.27.101.101.10.12.17.37.3.3.3.3 = Timeticks: (141000)
  0:23:30.00
iso.3.6.1.2.1.10.166.3.2.2.1.28.101.101.10.12.17.37.3.3.3.3 = Timeticks: (141000)
  0:23:30.00
iso.3.6.1.2.1.10.166.3.2.2.1.29.101.101.10.12.17.37.3.3.3.3 = Timeticks: (141000)
  0:23:30.00
iso.3.6.1.2.1.10.166.3.2.2.1.30.101.101.10.12.17.37.3.3.3.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.2.1.31.101.101.10.12.17.37.3.3.3.3 = Timeticks: (0)
  0:00:00.00
iso.3.6.1.2.1.10.166.3.2.2.1.32.101.101.10.12.17.37.3.3.3.3 = Timeticks: (6988400)
  19:24:44.00
iso.3.6.1.2.1.10.166.3.2.2.1.33.101.101.10.12.17.37.3.3.3.3 = INTEGER: 8
iso.3.6.1.2.1.10.166.3.2.2.1.34.101.101.10.12.17.37.3.3.3.3 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.2.1.35.101.101.10.12.17.37.3.3.3.3 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.2.1.36.101.101.10.12.17.37.3.3.3.3 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.2.1.37.101.101.10.12.17.37.3.3.3.3 = INTEGER: 2
```

`snmpget` on `mplsTunnelName` MIB object will return the `mplsTunnelName` from `mplsTunnelTable`

```
snmpget -v 2c -c test 10.12.17.37
  .1.3.6.1.2.1.10.166.3.2.2.1.5.101.101.10.12.17.37.3.3.3.3
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.2.1.5.101.101.10.12.17.37.3.3.3.3 = STRING: "Tnl-1"
```

`snmpgetnext` on `mplsTunnelName` MIB object will return the `mplsTunnelDescr` from `mplsTunnelTable`

```
snmpgetnext -v 2c -c test 10.12.17.37
  .1.3.6.1.2.1.10.166.3.2.2.1.5.101.101.10.12.17.37.3.3.3.3
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.2.1.6.101.101.10.12.17.37.3.3.3.3 = STRING: "PE-1 to PE-2"
```

SNMP WALK for mplsTunnelResourceTable

Note: If we want bandwidth reservation tunnel, configure mplsTunnelResourceMeanRate MIB object and then make mplsTunnelResourceRowStatus as active through SNMP.

`snmpwalk` on `mplsTunnelResourceTable` will returns entire values in `mplsTunnelResourceTable`

```
snmpwalk -v 2c -c test 10.12.17.37 .1.3.6.1.2.1.10.166.3.2.6.1
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.6.1.2.5 = INTEGER: 45000
```

```
iso.3.6.1.2.1.10.166.3.2.6.1.3.5 = INTEGER: 25000
iso.3.6.1.2.1.10.166.3.2.6.1.4.5 = INTEGER: 15000
iso.3.6.1.2.1.10.166.3.2.6.1.5.5 = INTEGER: 20000
iso.3.6.1.2.1.10.166.3.2.6.1.6.5 = INTEGER: 10000
iso.3.6.1.2.1.10.166.3.2.6.1.7.5 = INTEGER: 2
iso.3.6.1.2.1.10.166.3.2.6.1.8.5 = INTEGER: 55
iso.3.6.1.2.1.10.166.3.2.6.1.9.5 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.6.1.10.5 = INTEGER: 2
```

`snmpget` on mplsTunnelResourceMaxRate MIB object will return the mplsTunnelResourceMaxRate from mplsTunnelResourceTable.

```
snmpget -v 2c -c test 10.12.17.37 .1.3.6.1.2.1.10.166.3.2.6.1.2.5
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.6.1.2.5 = INTEGER: 45000
```

`snmpgetnext` on mplsTunnelResourceMaxRate MIB object will return the mplsTunnelResourceMeanRate from mplsTunnelResourceTable.

```
snmpgetnext -v 2c -c test 10.12.17.37 .1.3.6.1.2.1.10.166.3.2.6.1.2.5
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.6.1.3.5 = INTEGER: 25000
```

SNMP WALK for mplsTunnelHopTable

If tunnel has to traverse any explicit path other than existing routing path, configure following MIB objects of mplsTunnelHopTable and then make mplsTunnelHopRowStatus as active through SNMP.

- mplsTunnelHopIpAddr
- mplsTunnelHopIpPrefixLen
- mplsTunnelHopType
- mplsTunnelHopPathOptionName

`snmpwalk` on mplsTunnelHopTable will returns entire values in mplsTunnelHopTable.

```
snmpwalk -v 2c -c test 10.12.17.37 .1.3.6.1.2.1.10.166.3.2.4
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.4.1.4.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.4.1.5.1.1.1 = STRING: "0.0.0.0"
iso.3.6.1.2.1.10.166.3.2.4.1.6.1.1.1 = INTEGER: 32
iso.3.6.1.2.1.10.166.3.2.4.1.7.1.1.1 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.4.1.8.1.1.1 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.4.1.9.1.1.1 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.4.1.10.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.4.1.11.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.4.1.12.1.1.1 = STRING: "first"
```

```
iso.3.6.1.2.1.10.166.3.2.4.1.13.1.1.1 = INTEGER: 0
iso.3.6.1.2.1.10.166.3.2.4.1.14.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.10.166.3.2.4.1.15.1.1.1 = INTEGER: 2
```

`snmpget` on mplsTunnelHopInclude MIB object will return the mplsTunnelHopInclude from mplsTunnelHopTable.

```
snmpget -v 2c -c test 10.12.17.37 .1.3.6.1.2.1.10.166.3.2.4.1.11.1.1.1
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.4.1.11.1.1.1 = INTEGER: 1
```

`snmpgetnext` on mplsTunnelHopInclude MIB object will return the mplsTunnelHopPathOptionName from mplsTunnelHopTable.

```
snmpgetnext -v 2c -c test 10.12.17.37 .1.3.6.1.2.1.10.166.3.2.4.1.11.1.1.1
```

Sample output

```
iso.3.6.1.2.1.10.166.3.2.4.1.12.1.1.1 = STRING: "first"
```

Configuring LDP-MIB through SNMP

RFC3815 supports following tables

- mplsLdpEntityTable

Pre-requisite:

1. Pre-configure routing protocol OR static routes
2. Pre-configure label switching for interface
3. For creating LDP targeted session, convert the destination IP address into a decimal value. This value has to be passed as an index value for the LDP entity row creation and the same IP address needs to be configured for the mplsLdpEntityTargetPeer MIB object in mplsLdpEntityTable

SNMP WALK for mplsLdpEntityTable

snmpwalk on mplsLdpEntityTable will returns entire values in mplsLdpEntityTable

snmpwalk -v 2c -c test 10.12.17.39 .1.3.6.1.2.1.10.166.4.1.2.3.1

Sample output

```
iso.3.6.1.2.1.10.166.4.1.2.3.1.3.1.1.1.1.0.0.3 = Gauge32: 1
iso.3.6.1.2.1.10.166.4.1.2.3.1.4.1.1.1.1.0.0.3 = INTEGER: 1
iso.3.6.1.2.1.10.166.4.1.2.3.1.5.1.1.1.1.0.0.3 = INTEGER: 2
iso.3.6.1.2.1.10.166.4.1.2.3.1.6.1.1.1.1.0.0.3 = Gauge32: 646
iso.3.6.1.2.1.10.166.4.1.2.3.1.7.1.1.1.1.0.0.3 = Gauge32: 646
iso.3.6.1.2.1.10.166.4.1.2.3.1.8.1.1.1.1.0.0.3 = Gauge32: 4096
iso.3.6.1.2.1.10.166.4.1.2.3.1.9.1.1.1.1.0.0.3 = Gauge32: 30
iso.3.6.1.2.1.10.166.4.1.2.3.1.10.1.1.1.1.0.0.3 = Gauge32: 0
iso.3.6.1.2.1.10.166.4.1.2.3.1.11.1.1.1.1.0.0.3 = INTEGER: 8
iso.3.6.1.2.1.10.166.4.1.2.3.1.12.1.1.1.1.0.0.3 = INTEGER: 2
iso.3.6.1.2.1.10.166.4.1.2.3.1.13.1.1.1.1.0.0.3 = INTEGER: 2
iso.3.6.1.2.1.10.166.4.1.2.3.1.14.1.1.1.1.0.0.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.4.1.2.3.1.15.1.1.1.1.0.0.3 = INTEGER: 0
iso.3.6.1.2.1.10.166.4.1.2.3.1.16.1.1.1.1.0.0.3 = INTEGER: 2
iso.3.6.1.2.1.10.166.4.1.2.3.1.17.1.1.1.1.0.0.3 = INTEGER: 2
```

```
iso.3.6.1.2.1.10.166.4.1.2.3.1.18.1.1.1.1.0.0.3 = INTEGER: 1
iso.3.6.1.2.1.10.166.4.1.2.3.1.19.1.1.1.1.0.0.3 = IpAddress: 0.0.0.0
iso.3.6.1.2.1.10.166.4.1.2.3.1.20.1.1.1.1.0.0.3 = INTEGER: 1
iso.3.6.1.2.1.10.166.4.1.2.3.1.21.1.1.1.1.0.0.3 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.10.166.4.1.2.3.1.22.1.1.1.1.0.0.3 = INTEGER: 2
iso.3.6.1.2.1.10.166.4.1.2.3.1.23.1.1.1.1.0.0.3 = INTEGER: 1
```

snmpget on mplsLdpEntityAdminStatus MIB object will return the mplsLdpEntityAdminStatus from mplsLdpEntityTable

snmpget -v 2c -c test 10.12.17.39 .1.3.6.1.2.1.10.166.4.1.2.3.1.4.1.1.1.1.0.0.3

Sample output

```
iso.3.6.1.2.1.10.166.4.1.2.3.1.4.1.1.1.1.0.0.3 = INTEGER: 1
```

snmpgetnext on mplsLdpEntityAdminStatus MIB object will return the mplsLdpEntityOperStatus from mplsLdpEntityTable

```
snmpgetnext -v 2c -c test 10.12.17.39
.1.3.6.1.2.1.10.166.4.1.2.3.1.4.1.1.1.1.0.0.3
```

Sample output

```
iso.3.6.1.2.1.10.166.4.1.2.3.1.5.1.1.1.1.0.0.3 = INTEGER: 2
```

Appendix E Building ZebOS-XP for Wind River Linux

This appendix contains the procedures to configure the Wind River Linux 5 ZebOS-XP build environment.

Build Wind River Linux 5

The `sample_createenv.sh` script in `ZebOS5/kernel/linux/wrlinux-5.0` creates the build environment.

```
#!/bin/bash

# Defaults
export WRL_BASE=/opt/wrl-5
export BOARD=x86-kvm-guest
export ROOTFS=glibc_small

# Get location
pushd $(dirname "$0")
export WRL_TEMPLATE=`pwd`
popd

# Bail if not run inside of ZebOS source tree
echo $WRL_TEMPLATE | grep wrlinux-5 &>/dev/null
if [ $? -ne 0 ] ; then
    echo "Script must be run inside of source tree"
    exit 100
fi

# Setup environment
$WRL_BASE/wrenv.sh -p wrlinux-5 -f sh -o print_env > env.sh
. ./env.sh

# Configure target
$WIND_LINUX_CONFIGURE \
    --enable-board=$BOARD \
    --enable-kernel=standard \
    --enable-rootfs=$ROOTFS \
    "$@"

# Copy over ZebOS kernel template
tar cfB - -C $WRL_TEMPLATE/layers . | tar xfB - -C ./layers
```

Build ZebOS-XP

The `sample_setenv.sh` script in `ZebOS5/kernel/linux/wrlinux-5.0` sets up the generated environment to build ZebOS.

```
# Sample environment setup script.  Source from inside the WRL 5 build
environment

export TOP_BUILD_DIR=`pwd`
export KERNEL_OUT=$TOP_BUILD_DIR/build/linux-windriver-3.4-r0/linux-x86-kvm-
guest-standard-build
export KERNEL_SOURCE=$TOP_BUILD_DIR/build/linux-windriver-3.4-r0/linux

export JOBS="grep proc /proc/cpuinfo | wc -l"

export C_INCLUDE_PATH=$TOP_BUILD_DIR/header
export COMPILER_DIR=$TOP_BUILD_DIR/bitbake_build/tmp/sysroots/x86_64-linux/
usr/bin/toolchain
export COMPILER_PREFIX=i586-wrswrap-linux-gnu-
export PATH=$PATH:$COMPILER_DIR

[ ! -d header ] && mkdir -p header/linux
ln -sf $KERNEL_OUT/include/generated header
ln -sf $KERNEL_OUT/include/linux/version.h header/linux
```

Appendix F Copyright Information

Third-party Copyright and License Information

NET-SNMP Software

Copyright 1989, 1991, 1992 by Carnegie Mellon University Derivative Work - 1996, 1998-2000

Copyright 1996, 1998-2000 The Regents of the University of California All Rights Reserved

Copyright 2001-2008 SourceForge.net, All Rights Reserved

Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL CMU OR THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM THE LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 2001, Networks Associates Technology, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the NAI Labs nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of this code are copyright (c) 2001, Cambridge Broadband Ltd. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- The name of Cambridge Broadband Ltd. may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright Information for MD5 Authentication Software

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

Copyright (C) 1999-2004, RSA Data Security, Inc. All Rights Reserved

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

Copyright Information for Libedit BSD Software Library

Copyright (C) 1992, 1993 The Regents of the University of California. All rights reserved.

Copyright (C) 1994-2008 The NetBSD Foundation, Inc. All Rights Reserved

This code is derived from software contributed to Berkeley by Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

4. This product includes software developed by the University of California, Berkeley and its contributors.

5. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR

TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Linux Kernel Patches

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

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

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA or visit http://www.gnu.org/licenses/gpl.html.

In accordance with the GNU Public License, IP Infusion offers to make source code for these Linux Kernel patches available to the customer upon request for a charge no more than our cost of physically performing the distribution.

Index

E

I

L

M

O

P

R

S

T

V

Z