System Management Configuration Guide for Cisco NCS 6000 Series Routers, IOS XR Release 7.2.x

First Published: 2020-01-29

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CONTENTS

Preface

Note This product has reached end-of-life status. For more information, see the End-of-Life and End-of-Sale Notices.

This guide describes the System Management configuration details for Cisco IOS XR software. This chapter contains details on the changes made to this document.

Changes to This Document

This table lists the changes made to this document since it was first released.

Table 1: Changes to This Document

Date	Summary
August 2020	Initial release of this document.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

CHAPTER 1

New and Changed System Management Features

This chapter lists all the features that have been added or modified in this guide. The table also contains references to these feature documentation sections.

- System Management Features Added or Modified in IOS XR Release 7.2.x, on page 1

System Management Features Added or Modified in IOS XR Release 7.2.x

Feature	Description	Changed in Release	Where Documented
No new features introduced.	Not applicable	Not applicable	Not applicable

CHAPTER 2

Configuring Manageability

This module describes the configuration required to enable the Extensible Markup Language (XML) agent services. The XML Parser Infrastructure provides parsing and generation of XML documents with Document Object Model (DOM), Simple Application Programming Interface (API) for XML (SAX), and Document Type Definition (DTD) validation capabilities:

- DOM allows customers to programmatically create, manipulate, and generate XML documents.
- SAX supports user-defined functions for XML tags.
- DTD allows for validation of defined document types.

Table 2: Feature History for Configuring Manageability on Cisco IOS XR Software

Release 5.0.0	This feature was introduced.

This module contains the following topics:

Information About XML Manageability

The Cisco IOS XR Extensible Markup Language (XML) API provides a programmable interface to the router for use by external management applications. This interface provides a mechanism for router configuration and monitoring utilizing XML formatted request and response streams. The XML interface is built on top of the Management Data API (MDA), which provides a mechanism for Cisco IOS XR components to publish their data models through MDA schema definition files.

Cisco IOS XR software provides the ability to access the router via XML using a dedicated TCP connection, Secure Socket Layer (SSL), or a specific VPN routing and forwarding (VRF) instance.

How to Configure Manageability

Configuring the XML Agent

SUMMARY STEPS

1. **xml agent** [**ssl**]
2. **iteration on size** *iteration-size*
3. **session timeout** *timeout*
4. **throttle** {**memory** *size* | **process-rate** *tags*}
5. **vrf** { **default** | *vrf-name*} [**access-list** *access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**xml agent** [**ssl**] **Example:** `RP/0/RP0/CPU0:router:router(config)# xml agent`	Enables Extensible Markup Language (XML) requests over a dedicated TCP connection and enters XML agent configuration mode. Use the **ssl** keyword to enable XML requests over Secure Socket Layer (SSL).
Step 2	**iteration on size** *iteration-size* **Example:** `RP/0/RP0/CPU0:router:router(config-xml-agent)# iteration on size 500`	Configures the iteration size for large XML agent responses in KBytes. The default is 48.
Step 3	**session timeout** *timeout* **Example:** `RP/0/RP0/CPU0:router:router(config-xml-agent)# session timeout 5`	Configures an idle timeout for the XML agent in minutes. By default, there is no timeout.

Step 4	**throttle** {**memory** *size* | **process-rate** *tags*}
	Example:
	`RP/0/RP0/CPU0:router:router(config-xml-agent)# throttle memory 300`
		Configures the XML agent processing capabilities.
		• Specify the throttle memory size in Mbytes per session. Values can range from 100 to 600. The default is 300.
		• Specify the process-rate as the number of tags that the XML agent can process per second. Values can range from 1000 to 30000. By default the process rate is not throttled.

Step 5	**vrf** { **default** | *vrf-name*} [**access-list** *access-list-name*] **Example:** `RP/0/RP0/CPU0:router:router(config-xml-agent)# vrf my-vrf`	Configures the dedicated agent or SSL agent to receive and send messages via the specified VPN routing and forwarding (VRF) instance.

Configuration Examples for Manageability

Enabling VRF on an XML Agent: Examples

The following example illustrates how to configure the dedicated XML agent to receive and send messages via VRF1, VRF2 and the default VRF:

```
RP/0/RP0/CPU0:router:router(config)# xml agent
RP/0/RP0/CPU0:router:router(config-xml-agent)# vrf VRF1
RP/0/RP0/CPU0:router:router(config-xml-agent)# vrf VRF2
```

The following example illustrates how to remove access to VRF2 from the dedicated agent:

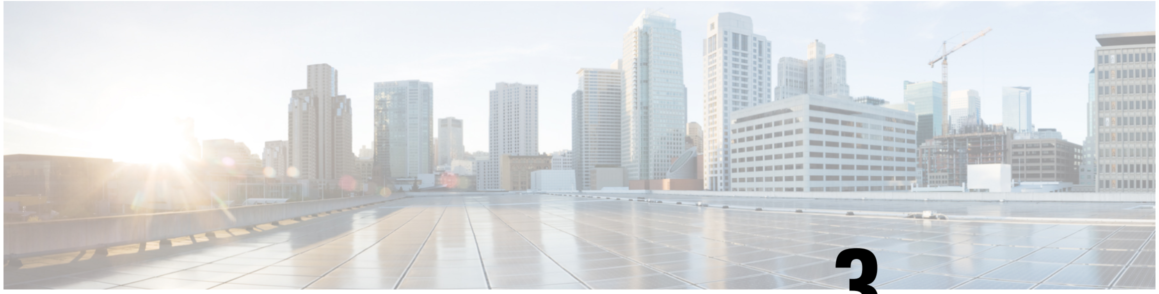
```
RP/0/RP0/CPU0:router:router(config)# xml agent
RP/0/RP0/CPU0:router:router(config-xml-agent)# no vrf VRF2
```

The following example shows how to configure the XML SSL agent to receive and send messages through VRF1, VRF2 and the default VRF:

```
RP/0/RP0/CPU0:router:router(config)# xml agent ssl
RP/0/RP0/CPU0:router:router(config-xml-agent)# vrf VRF1
RP/0/RP0/CPU0:router:router(config-xml-agent)# vrf VRF2
```

The following example removes access for VRF2 from the dedicated XML agent:

```
RP/0/RP0/CPU0:router:router(config)# xml agent ssl
RP/0/RP0/CPU0:router:router(config-xml-agent)# no vrf VRF2
```



CHAPTER 3

Configuring Physical and Virtual Terminals

Line templates define standard attribute settings for incoming and outgoing transport over physical and virtual terminal lines (vtys). Vty pools are used to apply template settings to ranges of vtys.

Note Before creating or modifying the vty pools, enable the telnet server using the **telnet server** command in XR Config mode. See *IP Addresses and Services Configuration Guide for Cisco NCS 6000 Series Routers* and *IP Addresses and Services Command Reference for Cisco NCS 6000 Series Routers* for more information.

This module describes the new and revised tasks you need to implement physical and virtual terminals on your Cisco IOS XR network.

For more information about physical and virtual terminals on the Cisco IOS XR software and complete descriptions of the terminal services commands listed in this module, see Related Documents, on page 17. To locate documentation for other commands that might appear in the course of running a configuration task, search online in .

Table 3: Feature History for Implementing Physical and Virtual Templates on Cisco IOS XR Software

Release	Modification
Release 5.0.0	This feature was introduced.

This module contains the following topics:

Prerequisites for Implementing Physical and Virtual Terminals

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Information About Implementing Physical and Virtual Terminals

To implement physical and virtual terminals, you need to understand the concepts in this section.

Tip You can programmatically manage the physical and virtual terminals using `openconfig-system-terminal.yang` OpenConfig data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco NCS 6000 Series Routers*.

Line Templates

The following line templates are available in the Cisco IOS XR software.

- Default line template—The default line template that applies to a physical and virtual terminal lines.
- Console line template—The line template that applies to the console line.
- User-defined line templates—User-defined line templates that can be applied to a range of virtual terminal lines.

Line Template Configuration Mode

Changes to line template attributes are made in line template configuration mode. To enter line template configuration mode, issue the **line** command from XR Config mode, specifying the template to be modified. These line templates can be configured with the **line** command:

- console—console template
- default—default template
- template—user-defined template

After you specify a template with the **line** command, the router enters line template configuration mode where you can set the terminal attributes for the specified line. This example shows how to specify the attributes for the console:

```
RP/0/RP0/CPU0:router(config)# line console
RP/0/RP0/CPU0:router(config-line)#
```

From line template configuration mode, use the online help feature (**?**) to view all available options. Some useful options include:

- absolute-timeout—Specifies a timeout value for line disconnection.
- escape-character—Changes the line escape character.
- exec-timeout—Specifies the EXEC timeout.
- length—Sets the number of lines displayed on the screen.
- session-limit—Specifies the allowable number of outgoing connections.

- session-timeout—Specifies an interval for closing the connection if there is no input traffic.
- timestamp—Displays the timestamp before each command.
- width—Specifies the width of the display terminal.

Note The *default* session-limit for line template is applicable to Telnet sessions only. It is not applicable for SSH sessions.

Line Template Guidelines

The following guidelines apply to modifying the console template and to configuring a user-defined template:

- Modify the templates for the physical terminal lines on the router (the console port) from line template configuration mode. Use the **line console** command from XR Config mode to enter line template configuration mode for the console template.
- Modify the template for virtual lines by configuring a user-defined template with the **line** *template-name* command, configuring the terminal attributes for the user-defined template from line template configuration, and applying the template to a range of virtual terminal lines using the **vty pool** command.

Attributes not defined in the console template, or any virtual template, are taken from the default template.

The default settings for the default template are described for all commands in line template configuration mode in the *Terminal Services Commands on* module in *System Management Command Reference for Cisco NCS 6000 Series Routers*.

Note Before creating or modifying the vty pools, enable the telnet server using the **telnet server** command in XR Config mode. See *IP Addresses and Services Configuration Guide for Cisco NCS 6000 Series Routers* and *IP Addresses and Services Command Reference for Cisco NCS 6000 Series Routers* for more information.

Terminal Identification

The physical terminal lines for the console port is identified by its location, expressed in the format of *rack/slot/module* , on the active or standby route processor (RP) where the respective console port resides. For virtual terminals, physical location is not applicable; the Cisco IOS XR software assigns a vty identifier to vtys according to the order in which the vty connection has been established.

vty Pools

Each virtual line is a member of a pool of connections using a common line template configuration. Multiple vty pools may exist, each containing a defined number of vtys as configured in the vty pool. The Cisco IOS XR software supports the following vty pools by default:

- Default vty pool—The default vty pool consists of five vtys (vtys 0 through 4) that each reference the default line template.

- Default fault manager pool—The default fault manager pool consists of six vtys (vtys 100 through 105) that each reference the default line template.

In addition to the default vty pool and default fault manager pool, you can also configure a user-defined vty pool that can reference the default template or a user-defined template.

When configuring vty pools, follow these guidelines:

- The vty range for the default vty pool must start at vty 0 and must contain a minimum of five vtys.
- The vty range from 0 through 99 can reference the default vty pool.
- The vty range from 5 through 99 can reference a user-defined vty pool.
- The vty range from 100 is reserved for the fault manager vty pool.
- The vty range for fault manager vty pools must start at vty 100 and must contain a minimum of six vtys.
- A vty can be a member of only one vty pool. A vty pool configuration will fail if the vty pool includes a vty that is already in another pool.
- If you attempt to remove an active vty from the active vty pool when configuring a vty pool, the configuration for that vty pool will fail.

How to Implement Physical and Virtual Terminals on Cisco IOS XR Software

Modifying Templates

This task explains how to modify the terminal attributes for the console and default line templates. The terminal attributes that you set will modify the template settings for the specified template.

SUMMARY STEPS

1. **configure**
2. **line** {**console** | **default**}
3. Configure the terminal attribute settings for the specified template using the commands in line template configuration mode.
4. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.

	Command or Action	Purpose

Step 2	**line** {**console** | **default**}
	Example:
	`RP/0/RP0/CPU0:router(config)# line console`
	or
	`RP/0/RP0/CPU0:router(config)# line default`
		Enters line template configuration mode for the specified line template.
		- **console** —Enters line template configuration mode for the console template.
		- **default** —Enters line template configuration mode for the default line template.

Step 3	Configure the terminal attribute settings for the specified template using the commands in line template configuration mode.	—

Step 4	Use one of the following commands:
	- **end**
	- **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-line)# end`
	or
	`RP/0/RP0/CPU0:router(config-line)# commit`
		Saves configuration changes.
		- When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Creating and Modifying vty Pools

This task explains how to create and modify vty pools.

You can omit Step 3, on page 12 to Step 5, on page 12 if you are configuring the default line template to reference a vty pool.

SUMMARY STEPS

1. **configure**
2. **telnet** {**ipv4** | **ipv6**} **server max-servers** *limit*
3. **line template** *template-name*

4. Configure the terminal attribute settings for the specified line template using the commands in line template configuration mode.
5. **exit**
6. **vty-pool** {**default** | *pool-name* | **eem**} *first-vty last-vty* [**line-template** {**default** | *template-name*}]
7. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**telnet** {**ipv4** | **ipv6**} **server max-servers** *limit* **Example:** `RP/0/RP0/CPU0:router(config)# telnet ipv4 server max-servers 10`	Specifies the number of allowable Telnet servers. Up to 100 Telnet servers are allowed. **Note** By default no Telnet servers are allowed. You must configure this command in order to enable the use of Telnet servers.
Step 3	**line template** *template-name* **Example:** `RP/0/RP0/CPU0:router(config)# line template 1`	Enters line template configuration mode for a user-defined template.
Step 4	Configure the terminal attribute settings for the specified line template using the commands in line template configuration mode.	—
Step 5	**exit** **Example:** `RP/0/RP0/CPU0:router(config-line)# exit`	Exits line template configuration mode and returns the router to global configuration mode.

Step 6	**vty-pool** {**default** | *pool-name* | **eem**} *first-vty last-vty* [**line-template** {**default** | *template-name*}]
	Example:
	`RP/0/RP0/CPU0:router(config)# vty-pool default 0 5 line-template default`
	or
	`RP/0/RP0/CPU0:router(config)# vty-pool pool1 5 50 line-template template1`
	or
		Creates or modifies vty pools.
		• If you do not specify a line template with the **line-template** keyword, a vty pool defaults to the default line template.
		• **default** —Configures the default vty pool.
		  • The default vty pool must start at vty 0 and must contain a minimum of five vtys (vtys 0 through 4).
		  • You can resize the default vty pool by increasing the range of vtys that compose the default vty pool.

	Command or Action	Purpose

	`RP/0/RP0/CPU0:router(config)# vty-pool`
	`    eem 100 105 line-template template1`
		• *pool-name* —Creates a user-defined vty pool.
		  • A user-defined pool must start at least at vty 5, depending on whether the default vty pool has been resized.
		  • If the range of vtys for the default vty pool has been resized, use the first range value free from the default line template. For example, if the range of vtys for the default vty pool has been configured to include 10 vtys (vty 0 through 9), the range value for the user-defined vty pool must start with vty 10.
		• **eem** —Configures the embedded event manager pool.
		  • The default embedded event manager vty pool must start at vty 100 and must contain a minimum of six vtys (vtys 100 through 105).
		• **line-template** *template-name* —Configures the vty pool to reference a user-defined template.

Step 7	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Monitoring Terminals and Terminal Sessions

This task explains how to monitor terminals and terminal sessions using the **show** EXEC commands available for physical and terminal lines.

Note The commands can be entered in any order.

SUMMARY STEPS

1. (Optional) **show line** [**aux location** *node-id* | **console location** *node-id* | **vty** *number*]
2. (Optional) **show terminal**

3. (Optional) **show users**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **show line** [**aux location** *node-id* | **console location** *node-id* | **vty** *number*] **Example:** `RP/0/RP0/CPU0:router# show line`	Displays the terminal parameters of terminal lines. • Specifying the **show line aux location** *node-id* EXEC command displays the terminal parameters of the auxiliary line. • Specifying the **show line console location** *node-id* EXEC command displays the terminal parameters of the console. • For the **location** *node-id* keyword and argument, enter the location of the Route Processor (RP) on which the respective auxiliary or console port resides. • The *node-id* argument is expressed in the format of *rack*/*slot*/*module* . • Specifying the **show line vty** *number* EXEC command displays the terminal parameters for the specified vty.
Step 2	(Optional) **show terminal** **Example:** `RP/0/RP0/CPU0:router# show terminal`	Displays the terminal attribute settings for the current terminal line.
Step 3	(Optional) **show users** **Example:** `RP/0/RP0/CPU0:router# show users`	Displays information about the active lines on the router.

Craft Panel Interface

The Craft Panel is an easily-accessible and user-friendly interface which assists the field operator in troubleshooting the router. It consists of a LCD display and three LEDs. The LEDs indicate minor, major and critical alarms.

For more details of the Craft Panel Interface, refer the *Hardware and System set-up guides*.

Configuration Examples for Implementing Physical and Virtual Terminals

Modifying the Console Template: Example

This configuration example shows how to modify the terminal attribute settings for the console line template:

```
line console
  exec-timeout 0 0
  escape-character 0x5a
  session-limit 10
  disconnect-character 0x59
  session-timeout 100
  transport input telnet
  transport output telnet
```

In this configuration example, the following terminal attributes are applied to the console line template:

- The EXEC time out for terminal sessions is set to 0 minutes, 0 seconds. Setting the EXEC timeout to 0 minutes and 0 seconds disables the EXEC timeout function; thus, the EXEC session for the terminal session will never time out.
- The escape character is set to the 0x5a hexadecimal value (the 0x5a hexadecimal value translates into the “Z” character).
- The session limit for outgoing terminal sessions is set to 10 connections.
- The disconnect character is set to 0x59 hexadecimal value (the 0x59 hexadecimal character translates into the “Y” character).
- The session time out for outgoing terminal sessions is set to 100 minutes (1 hour and 40 minutes).
- The allowed transport protocol for incoming terminal sessions is Telnet.
- The allowed transport protocol for outgoing terminal sessions is Telnet.

To verify that the terminal attributes for the console line template have been applied to the console, use the **show line** command:

```
RP/0/RP0/CPU0:router# show line console location 0/0/CPU0

Tty             Speed     Modem  Uses   Noise Overruns           Acc I/O
*  con0/0/CPU0      9600        -     -       -      0/0               -/-

Line con0_0_CPU0, Location "Unknown", Type "Unknown"
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600, 1 parity, 2 stopbits, 8 databits
Template: console
Config:
Allowed transports are telnet.
```

Modifying the Default Template: Example

This configuration example shows how to override the terminal settings for the default line template:

```
line default
  exec-timeout 0 0
  width 512
  length 512
```

In this example, the following terminal attributes override the default line template default terminal attribute settings:

- The EXEC timeout for terminal sessions is set to 0 minutes and 0 seconds. Setting the EXEC timeout to 0 minutes and 0 seconds disables the EXEC timeout function; thus, the EXEC session for the terminal session will never time out (the default EXEC timeout for the default line template is 10 minutes).
- The width of the terminal screen for the terminals referencing the default template is set to 512 characters (the default width for the default line template is 80 characters).
- The length, the number of lines that will display at one time on the terminal referencing the default template, is set to 512 lines (the default length for the default line template is 24 lines).

Configuring a User-Defined Template to Reference the Default vty Pool: Example

This configuration example shows how to configure a user-defined line template (named test in this example) for vtys and to configure the line template test to reference the default vty pool:

```
line template test
  exec-timeout 100 0
  width 100
  length 100
  exit
vty-pool default 0 4 line-template test
```

Configuring a User-Defined Template to Reference a User-Defined vty Pool: Example

This configuration example shows how to configure a user-defined line template (named test2 in this example) for vtys and to configure the line template test to reference a user-defined vty pool (named pool1 in this example):

```
line template test2
  exec-timeout 0 0
  session-limit 10
  session-timeout 100
  transport input all
  transport output all
  exit
vty-pool pool1 5 50 line-template test2
```

Configuring a User-Defined Template to Reference the Fault Manager vty Pool: Example

This configuration example shows how to configure a user-defined line template (named test3 in this example) for vtys and to configure the line template test to reference the fault manager vty pool:

```
line template test3
  width 110
  length 100
  session-timeout 100
```

```
exit
vty-pool eem 100 106 line-template test3
```

Additional References

The following sections provide references related to implementing physical and virtual terminals on Cisco IOS XR software.

Related Documents

Related Topic	Document Title
Cisco IOS XR terminal services commands	*Terminal Services Commands on* module of *System Management Command Reference for Cisco NCS 6000 Series Routers*
Cisco IOS XR command master index	
Information about getting started with Cisco IOS XR software	
Information about user groups and task IDs	*Configuring AAA Services on* module of *System Security Configuration Guide for Cisco NCS 6000 Series Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 4

Configuring Simple Network Management Protocol

Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

This module describes the new and revised tasks you need to implement SNMP on your Cisco IOS XR network.

For detailed conceptual information about SNMP on the Cisco IOS XR software and complete descriptions of the SNMP commands listed in this module, see Related Documents, on page 45. For information on specific MIBs, refer to . To locate documentation for other commands that might appear in the course of performing a configuration task, search online in .

Table 4: Feature History for Implementing SNMP on Cisco IOS XR Software

Release	Modification
Release 3.9.0	Support was added for 3DES and AES encryption. The ability to preserve ENTITY-MIB and CISCO-CLASS-BASED-QOS-MIB data was added.
Release 4.2.0	Support was added for SNMP over IPv6.

This module contains the following topics:

Prerequisites for Implementing SNMP

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Restrictions for SNMP Use on Cisco IOS XR Software

SNMP outputs are only 32-bits wide and therefore cannot display any information greater than 2^{32}. 2^{32} is equal to 4.29 Gigabits. Note that a 10 Gigabit interface is greater than this and so if you are trying to display speed information regarding the interface, you might see concatenated results.

The recommended maximum number of object identifiers (OIDs) that can be accommodated in a single SNMP request is 75. A request with more than 75 OIDs can result in SNMP requests being dropped with SNMP polling timeout.

Information About Implementing SNMP

To implement SNMP, you need to understand the concepts described in this section.

SNMP Functional Overview

The SNMP framework consists of three parts:

- SNMP manager
- SNMP agent
- Management Information Base (MIB)

SNMP Manager

The SNMP manager is the system used to control and monitor the activities of network hosts using SNMP. The most common managing system is called a *network management system* (NMS). The term NMS can be applied to either a dedicated device used for network management, or the applications used on such a device. A variety of network management applications are available for use with SNMP. These features range from simple command-line applications to feature-rich graphical user interfaces (such as the CiscoWorks 2000 line of products).

SNMP Agent

The SNMP agent is the software component within the managed device that maintains the data for the device and reports these data, as needed, to managing systems. The agent and MIB reside on the router. To enable the SNMP agent, you must define the relationship between the manager and the agent.

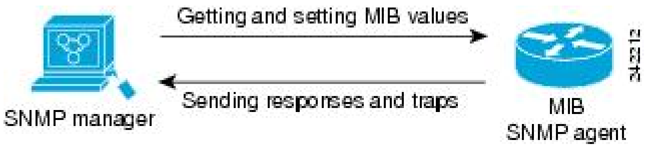
MIB

The *Management Information Base* (MIB) is a virtual information storage area for network management information, which consists of collections of managed objects. Within the MIB there are collections of related objects, defined in MIB modules. MIB modules are written in the SNMP MIB module language, as defined in STD 58, RFC 2578, RFC 2579, and RFC 2580. Note that individual MIB modules are also referred to as MIBs; for example, the Interfaces Group MIB (IF-MIB) is a MIB module within the MIB on your system.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change through Get or Set operations. A manager can get a value from an agent or store a value into that agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to manager requests to get or set data.

This figure illustrates the communications relationship between the SNMP manager and agent. A manager can send the agent requests to get and set MIB values. The agent can respond to these requests. Independent of this interaction, the agent can send unsolicited notifications (traps) to the manager to notify the manager of network conditions.

Figure 1: Communication Between an SNMP Agent and Manager



IP-MIB Support

RFC4293 IP-MIB was specifically designed to provide IPv4 and IPv6 statistics individually. The **ipIfStatsTable** defined in RFC 4293, lists the interface specific statistics. IPv6 statistics support in ipIfStatsTable was added earlier but, IOS-XR implementation of IP-MIB did not support IPv4 statistics as per RFC4293 in earlier releases.

From Release 6.3.2 onwards, IOS-XR implementation of IP-MIB supports IPv4 statistics as per RFC4293. This will enable you to collect the IPV4 and IPv6 statistics separately for each interface. The **ipIfStatsTable** is indexed by two **sub-ids address type (IPv4 or IPv6)** and the **interface ifindex[1]**. The implementation of IP-MIB support for IPv4 and IPv6 is separated from Release 6.3.2 for better readability and maintainability.

The list of OIDs added to the **ipIfStatsTable** for IPv4 statistics are:

- ipIfStatsInReceives
- ipIfStatsHCInReceives
- ipIfStatsInOctets
- ipIfStatsHCInOctets
- ipIfStatsOutTransmits
- ipIfStatsHCOutTransmits
- ipIfStatsOutOctets
- ipIfStatsHCOutOctets
- ipIfStatsDiscontinuityTime

For more information on the list of new OIDs added for iPv4 statistics, see SNMP OID Navigator.

Related Topics

Additional References, on page 45

SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. On Cisco IOS XR software, unsolicited (asynchronous) notifications can be generated only as *traps*. Traps are messages alerting the SNMP manager to a condition on the network. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

Note Inform requests (inform operations) are supported in Cisco IOS XR software from release 4.1 onwards. For more information see, http://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k_r5-3/sysman/command/reference/b-sysman-cr53xasr/b-sysman-cr53xasr_chapter_010010.html#wp2863682680

Traps are less reliable than informs because the receiver does not send any acknowledgment when it receives a trap. The sender cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the manager does not receive an inform request, it does not send a response. If the sender never receives a response, the inform request can be sent again. Thus, informs are more likely to reach their intended destination.

However, traps are often preferred because informs consume more resources in the router and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in memory until a response is received or the request times out. Also, traps are sent only once, and an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network. Thus, traps and inform requests provide a trade-off between reliability and resources.

Figure 2: Trap Received by the SNMP Manager

In this illustration, the agent router sends a trap to the SNMP manager. Although the manager receives the trap, it does not send any acknowledgment to the agent. The agent has no way of knowing that the trap reached its destination.

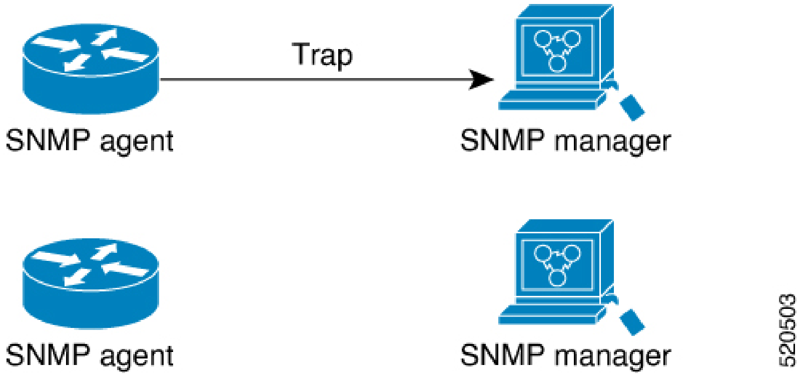
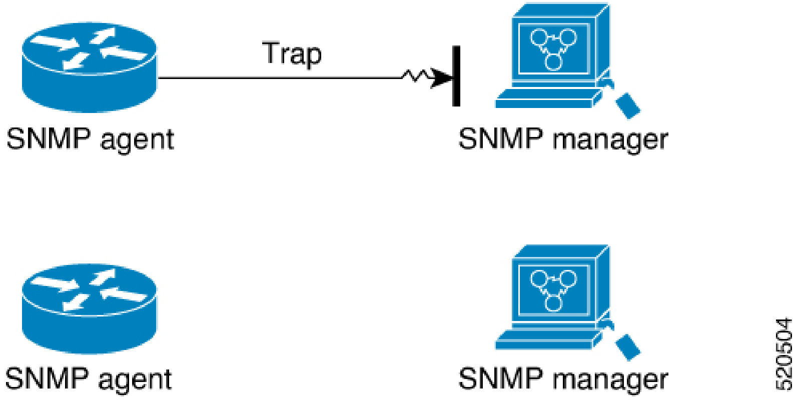


Figure 3: Trap Not Received by the SNMP Manager

In this illustration, the agent sends a trap to the manager, but the trap does not reach the manager. Because the agent has no way of knowing that the trap did not reach its destination, the trap is not sent again. The manager never receives the trap.



SNMP Versions

Cisco IOS XR software supports the following versions of SNMP:

- Simple Network Management Protocol Version 1 (SNMPv1)
- Simple Network Management Protocol Version 2c (SNMPv2c)
- Simple Network Management Protocol Version 3 (SNMPv3)

Both SNMPv1 and SNMPv2c use a community-based form of security. The community of managers able to access the agent MIB is defined by an IP address access control list and password.

SNMPv2c support includes a bulk retrieval mechanism and more detailed error message reporting to management stations. The bulk retrieval mechanism supports the retrieval of tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2c improved error handling support includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes now report the error type. Three kinds of exceptions are also reported: no such object exceptions, no such instance exceptions, and end of MIB view exceptions.

SNMPv3 is a security model. A *security model* is an authentication strategy that is set up for a user and the group in which the user resides. A *security level* is the permitted level of security within a security model. A combination of a security model and a security level will determine which security mechanism is employed when an SNMP packet is handled. See Table 6: SNMP Security Models and Levels, on page 24 for a list of security levels available in SNMPv3. The SNMPv3 feature supports RFCs 3411 to 3418.

You must configure the SNMP agent to use the version of SNMP supported by the management station. An agent can communicate with multiple managers; for this reason, you can configure the Cisco IOS-XR software to support communications with one management station using the SNMPv1 protocol, one using the SNMPv2c protocol, and another using SMNPv3.

Comparison of SNMPv1, v2c, and v3

SNMP v1, v2c, and v3 all support the following operations:

- get-request—Retrieves a value from a specific variable.

- get-next-request—Retrieves the value following the named variable; this operation is often used to retrieve variables from within a table. With this operation, an SNMP manager does not need to know the exact variable name. The SNMP manager searches sequentially to find the needed variable from within the MIB.
- get-response—Operation that replies to a get-request, get-next-request, and set-request sent by an NMS.
- set-request—Operation that stores a value in a specific variable.
- trap—Unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

The below table identifies other key SNMP features supported by the SNMP v1, v2c, and v3.

Table 5: SNMPv1, v2c, and v3 Feature Support

Feature	SNMP v1	SNMP v2c	SNMP v3
Get-Bulk Operation	No	Yes	Yes
Inform Operation	No	Yes (No on the Cisco IOS XR software)	Yes (No on the Cisco IOS XR software)
64 Bit Counter	No	Yes	Yes
Textual Conventions	No	Yes	Yes
Authentication	No	No	Yes
Privacy (Encryption)	No	No	Yes
Authorization and Access Controls (Views)	No	No	Yes

Security Models and Levels for SNMPv1, v2, v3

The security level determines if an SNMP message needs to be protected from disclosure and if the message needs to be authenticated. The various security levels that exist within a security model are as follows:

- noAuthNoPriv—Security level that does not provide authentication or encryption.
- authNoPriv—Security level that provides authentication but does not provide encryption.
- authPriv—Security level that provides both authentication and encryption.

Three security models are available: SNMPv1, SNMPv2c, and SNMPv3. The security model combined with the security level determine the security mechanism applied when the SNMP message is processed.

The below table identifies what the combinations of security models and levels mean.

Table 6: SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.

Model	Level	Authentication	Encryption	What Happens
v2c	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	HMAC-MD5 or HMAC-SHA	No	Provides authentication based on the HMAC[1]-MD5[2] algorithm or the HMAC-SHA[3].
v3	authPriv	HMAC-MD5 or HMAC-SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES[4] 56-bit encryption in addition to authentication based on the CBC[5] DES (DES-56) standard.
v3	authPriv	HMAC-MD5 or HMAC-SHA	3DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides 168-bit 3DES[6] level of encryption.
v3	authPriv	HMAC-MD5 or HMAC-SHA	AES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides 128-bit AES[7] level of encryption.

[1] Hash-Based Message Authentication Code
[2] Message Digest 5
[3] Secure Hash Algorithm
[4] Data Encryption Standard
[5] Cipher Block Chaining
[6] Triple Data Encryption Standard
[7] Advanced Encryption Standard

Use of 3DES and AES encryption standards requires that the security package (k9sec) be installed. For information on installing software packages, see *Upgrading and Managing Cisco IOS XR Software*.

SNMPv3 Benefits

SNMPv3 provides secure access to devices by providing authentication, encryption and access control. These added security benefits secure SNMP against the following security threats:

- Masquerade—The threat that an SNMP user may assume the identity of another SNMP user to perform management operations for which that SNMP user does not have authorization.
- Message stream modification—The threat that messages may be maliciously reordered, delayed, or replayed (to an extent that is greater than can occur through the natural operation of a subnetwork service) to cause SNMP to perform unauthorized management operations.
- Disclosure—The threat that exchanges between SNMP engines could be eavesdropped. Protecting against this threat may be required as a matter of local policy.

In addition, SNMPv3 provides access control over protocol operations on SNMP managed objects.

SNMPv3 Costs

SNMPv3 authentication and encryption contribute to a slight increase in the response time when SNMP operations on MIB objects are performed. This cost is far outweighed by the security advantages provided by SNMPv3.

This table shows the order of response time (from least to greatest) for the various security model and security level combinations.

Table 7: Order of Response Times from Least to Greatest

Security Model	Security Level
SNMPv2c	noAuthNoPriv
SNMPv3	noAuthNoPriv
SNMPv3	authNoPriv
SNMPv3	authPriv

User-Based Security Model

SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity—Ensures that messages have not been altered or destroyed in an unauthorized manner and that data sequences have not been altered to an extent greater than can occur nonmaliciously.
- Message origin authentication—Ensures that the claimed identity of the user on whose behalf received data was originated is confirmed.
- Message confidentiality—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMPv3 authorizes management operations only by configured users and encrypts SNMP messages.

USM uses two authentication protocols:

- HMAC-MD5-96 authentication protocol
- HMAC-SHA-96 authentication protocol

USM uses Cipher Block Chaining (CBC)-DES (DES-56) as the privacy protocol for message encryption.

View-Based Access Control Model

The View-Based Access Control Model (VACM) enables SNMP users to control access to SNMP managed objects by supplying read, write, or notify access to SNMP objects. It prevents access to objects restricted by views. These access policies can be set when user groups are configured with the **snmp-server group** command.

MIB Views

For security reasons, it is often valuable to be able to restrict the access rights of some groups to only a subset of the management information within the management domain. To provide this capability, access to a management object is controlled through MIB views, which contain the set of managed object types (and, optionally, the specific instances of object types) that can be viewed.

Access Policy

Access policy determines the access rights of a group. The three types of access rights are as follows:

- read-view access—The set of object instances authorized for the group when objects are read.
- write-view access—The set of object instances authorized for the group when objects are written.
- notify-view access—The set of object instances authorized for the group when objects are sent in a notification.

IP Precedence and DSCP Support for SNMP

SNMP IP Precedence and differentiated services code point (DSCP) support delivers QoS specifically for SNMP traffic. You can change the priority setting so that SNMP traffic generated in a router is assigned a specific QoS class. The IP Precedence or IP DSCP code point value is used to determine how packets are handled in weighted random early detection (WRED).

After the IP Precedence or DSCP is set for the SNMP traffic generated in a router, different QoS classes cannot be assigned to different types of SNMP traffic in that router.

The IP Precedence value is the first three bits in the type of service (ToS) byte of an IP header. The IP DSCP code point value is the first six bits of the differentiate services (DiffServ Field) byte. You can configure up to eight different IP Precedence markings or 64 different IP DSCP markings.

How to Implement SNMP on Cisco IOS XR Software

This section describes how to implement SNMP.

The **snmp-server** commands enable SNMP on Management Ethernet interfaces by default. For information on how to enable SNMP server support on other inband interfaces, see the *Implementing Management Plane Protection on Cisco IOS XR Software* module in *System Security Configuration Guide for Cisco NCS 6000 Series Routers*.

Configuring SNMPv3

This task explains how to configure SNMPv3 for network management and monitoring.

Note No specific command enables SNMPv3; the first **snmp-server** global configuration command (config), that you issue enables SNMPv3. Therefore, the sequence in which you issue the **snmp-server** commands for this task does not matter.

SUMMARY STEPS

1. **configure**
2. **snmp-server view** *view-name oid-tree* {**included** | **excluded**}

3. **snmp-server group** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*]
4. **snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**priv des56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*]
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.

Step 2	**snmp-server view** *view-name oid-tree* {**included** | **excluded**}
	Example:
	`RP/0/RP0/CPU0:router(config)# snmp-server view view_name 1.3.6.1.2.1.1.5 included`
		Creates or modifies a view record.

Step 3	**snmp-server group** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server group group_name v3 noauth read view_name1 write view_name2`	Configures a new SNMP group or a table that maps SNMP users to SNMP views.
Step 4	**snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**priv des56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server user noauthuser group_name v3`	Configures a new user to an SNMP group. **Note** Only one remote host can be assigned to the same username for SNMP version 3. If you configure the same username with different remote hosts, only the last username and remote host combination will be accepted and will be seen in the **show running** configuration. In the case of multiple SNMP managers, multiple unique usernames are required.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring SNMP Trap Notifications

This task explains how to configure the router to send SNMP trap notifications.

Note You can omit Step 3, on page 28 if you have already completed the steps documented under the Configuring SNMPv3, on page 27 task.

SUMMARY STEPS

1. **configure**
2. **snmp-server group** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*]
3. **snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**priv des56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*]
4. **snmp-server host** *address* [**traps**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
5. **snmp-server traps** [*notification-type*]
6. Use the **commit** or **end** command.
7. (Optional) **show snmp host**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**snmp-server group** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server group group_name v3 noauth read view_name1 write view_name2`	Configures a new SNMP group or a table that maps SNMP users to SNMP views.
Step 3	**snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**}	Configures a new user to an SNMP group.

	Command or Action	Purpose
	auth-password [**priv des56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server user noauthuser group_name v3`	**Note** Only one remote host can be assigned to the same username for SNMP version 3. If you configure the same username with different remote hosts, only the last username and remote host combination will be accepted and will be seen in the **show running** configuration. In the case of multiple SNMP managers, multiple unique usernames are required.
Step 4	**snmp-server host** *address* [**traps**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server host 12.26.25.61 traps version 3 noauth userV3noauth`	Specifies SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the notifications.
Step 5	**snmp-server traps** [*notification-type*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server traps bgp`	Enables the sending of trap notifications and specifies the type of trap notifications to be sent. • If a trap is not specified with the *notification-type* argument, all supported trap notifications are enabled on the router. To display which trap notifications are available on your router, enter the **snmp-server traps ?** command.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 7	(Optional) **show snmp host** **Example:** `RP/0/RP0/CPU0:router# show snmp host`	Displays information about the configured SNMP notification recipient (host), port number, and security model.

Setting the Contact, Location, and Serial Number of the SNMP Agent

This task explains how to set the system contact string, system location string, and system serial number of the SNMP agent.

Note The sequence in which you issue the **snmp-server** commands for this task does not matter.

SUMMARY STEPS

1. **configure**
2. (Optional) **snmp-server contact** *system-contact-string*
3. (Optional) **snmp-server location** *system-location*
4. (Optional) **snmp-server chassis-id** *serial-number*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.

Step 2	(Optional) **snmp-server contact** *system-contact-string*
	Example:
	`RP/0/RP0/CPU0:router(config)# snmp-server contact`
	`Dial System Operator at beeper # 27345`
		Sets the system contact string.

Step 3	(Optional) **snmp-server location** *system-location*
	Example:
	`RP/0/RP0/CPU0:router(config)# snmp-server location`
	`Building 3/Room 214`
		Sets the system location string.

Step 4	(Optional) **snmp-server chassis-id** *serial-number*
	Example:
	`RP/0/RP0/CPU0:router(config)# snmp-server`
	`chassis-id 1234456`
		Sets the system serial number.

Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		• **Cancel** —Remains in the configuration session, without committing the configuration changes.

Defining the Maximum SNMP Agent Packet Size

This task shows how to configure the largest SNMP packet size permitted when the SNMP server is receiving a request or generating a reply.

Note The sequence in which you issue the **snmp-server** commands for this task does not matter.

SUMMARY STEPS

1. **configure**
2. (Optional) **snmp-server packetsize** *byte-count*
3. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	(Optional) **snmp-server packetsize** *byte-count* **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server packetsize 1024`	Sets the maximum packet size.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Changing Notification Operation Values

After SNMP notifications have been enabled, you can specify a value other than the default for the source interface, message queue length, or retransmission interval.

This task explains how to specify a source interface for trap notifications, the message queue length for each host, and the retransmission interval.

Note The sequence in which you issue the **snmp-server** commands for this task does not matter.

SUMMARY STEPS

1. **configure**
2. (Optional) **snmp-server trap-source** *type interface-path-id*
3. (Optional) **snmp-server queue-length** *length*
4. (Optional) **snmp-server trap-timeout** *seconds*
5. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	(Optional) **snmp-server trap-source** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server trap-source POS 0/0/1/0`	Specifies a source interface for trap notifications.
Step 3	(Optional) **snmp-server queue-length** *length* **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server queue-length 20`	Establishes the message queue length for each notification.
Step 4	(Optional) **snmp-server trap-timeout** *seconds* **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server trap-timeout 20`	Defines how often to resend notifications on the retransmission queue.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

	Command or Action	Purpose
		• **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Setting IP Precedence and DSCP Values

This task describes how to configure IP Precedence or IP DSCP for SNMP traffic.

Before you begin

SNMP must be configured.

SUMMARY STEPS

1. **configure**
2. Use one of the following commands:
   - **snmp-server ipv4 precedence** *value*
   - **snmp-server ipv4 dscp** *value*
3. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	Use one of the following commands: • **snmp-server ipv4 precedence** *value* • **snmp-server ipv4 dscp** *value* **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server dscp 24`	Configures an IP precedence or IP DSCP value for SNMP traffic.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session.

	Command or Action	Purpose
		• **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring MIB Data to be Persistent

Many SNMP MIB definitions define arbitrary 32-bit indices for their object tables. MIB implementations often do a mapping from the MIB indices to some internal data structure that is keyed by some other set of data. In these MIB tables the data contained in the table are often other identifiers of the element being modelled. For example, in the ENTITY-MIB, entries in the entPhysicalTable are indexed by the 31-bit value, entPhysicalIndex, but the entities could also be identified by the entPhysicalName or a combination of the other objects in the table.

Because of the size of some MIB tables, significant processing is required to discover all the mappings from the 32-bit MIB indices to the other data which the network management station identifies the entry. For this reason, it may be necessary for some MIB indices to be persistent across process restarts, switchovers, or device reloads. The ENTITY-MIB entPhysicalTable and CISCO-CLASS-BASED-QOS-MIB are two such MIBs that often require index values to be persistent.

Also, because of query response times and CPU utilization during CISCO-CLASS-BASED-QOS-MIB statistics queries, it is desirable to cache service policy statistics.

SUMMARY STEPS

1. (Optional) **snmp-server entityindex persist**
2. (Optional) **snmp-server mibs cbqosmib persist**
3. (Optional) **snmp-server cbqosmib cache refresh time** *time*
4. (Optional) **snmp-server cbqosmib cache service-policy count** *count*
5. **snmp-server ifindex persist**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **snmp-server entityindex persist** **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server entityindex persist`	Enables the persistent storage of ENTITY-MIB data.
Step 2	(Optional) **snmp-server mibs cbqosmib persist** **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib persist`	Enables persistent storage of the CISCO-CLASS-BASED-QOS-MIB data.

	Command or Action	Purpose
Step 3	(Optional) **snmp-server cbqosmib cache refresh time** *time* **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib cache refresh time 45`	Enables QoS MIB caching with a specified cache refresh time.
Step 4	(Optional) **snmp-server cbqosmib cache service-policy count** *count* **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server mibs cbqosmib cache service-policy count 50`	Enables QoS MIB caching with a limited number of service policies to cache.
Step 5	**snmp-server ifindex persist** **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server ifindex persist`	Enables ifIndex persistence globally on all Simple Network Management Protocol (SNMP) interfaces.

Configuring LinkUp and LinkDown Traps for a Subset of Interfaces

By specifying a regular expression to represent the interfaces for which you are interested in setting traps, you can enable or disable linkUp and linkDown traps for a large number of interfaces simultaneously.

Before you begin

SNMP must be configured.

SUMMARY STEPS

1. **configure**
2. **snmp-server interface subset** *subset-number* **regular-expression** *expression*
3. **notification linkupdown disable**
4. Use the **commit** or **end** command.
5. (Optional) **show snmp interface notification subset** *subset-number*
6. (Optional) **show snmp interface notification regular-expression** *expression*
7. (Optional) **show snmp interface notification** *type interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:**	Enters XR Config mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	

Step 2	**snmp-server interface subset** *subset-number* **regular-expression** *expression*
	Example:
	`RP/0/RP0/CPU0:router(config)# snmp-server interface`
	` subset 10`
	`    regular-expression "^Gig[a-zA-Z]+[0-9/]+\."`
	`RP/0/RP0/CPU0:router(config-snmp-if-subset)#`
		Enters snmp-server interface mode for the interfaces identified by the regular expression.
		The subset-number argument identifies the set of interfaces, and also assigns a priority to the subset in the event that an interface is included in more than one subset. Lower numbers have higher priority and their configuration takes precedent over interface subsets with higher numbers.
		The *expression* argument must be entered surrounded by double quotes.
		Refer to the *Understanding Regular Expressions, Special Characters, and Patterns* module in for more information regarding regular expressions.

Step 3	**notification linkupdown disable**
	Example:
	`RP/0/RP0/CPU0:router(config-snmp-if-subset)#`
	`notification linkupdown disable`
		Disables linkUp and linkDown traps for all interfaces being configured. To enable previously disabled interfaces, use the **no** form of this command.

Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes, and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration mode, without committing the configuration changes.

Step 5	(Optional) **show snmp interface notification subset** *subset-number*
	Example:
	`RP/0/RP0/CPU0:router# show snmp interface`
	`notification subset 10`
		Displays the linkUp and linkDown notification status for all interfaces identified by the subset priority.

Step 6	(Optional) **show snmp interface notification regular-expression** *expression*
	Example:
	`RP/0/RP0/CPU0:router# show snmp interface`
	`notification`
	`    regular-expression "^Gig[a-zA-Z]+[0-9/]+\."`
		Displays the linkUp and linkDown notification status for all interfaces identified by the regular expression.

	Command or Action	Purpose

Step 7	(Optional) **show snmp interface notification** *type interface-path-id*
	Example:
	`RP/0/RP0/CPU0:router# show snmp interface notification`
	`    tengige 0/4/0/3.10`
		Displays the linkUp and linkDown notification status for the specified interface.

Configuration Examples for Implementing SNMP

Configuring SNMPv3: Examples

Setting an Engine ID

This example shows how to set the identification of the local SNMP engine:

```
snmp-server engineID local 00:00:00:09:00:00:00:a1:61:6c:20:61
```

Note After the engine ID has been configured, the SNMP agent restarts.

Verifying the Identification of the Local SNMP Engines

This example shows how to verify the identification of the local SNMP engine:

```
config
  show snmp engineid

  SNMP engineID 00000009000000a1ffffffff
```

Creating a View

There are two ways to create a view:

- You can include the object identifier (OID) of an ASN.1 subtree of a MIB family from a view by using the **included** keyword of the **snmp-server view** command.
- You can exclude the OID subtree of the ASN.1 subtree of a MIB family from a view by using the **excluded** keyword of the **snmp-server view** command.

This example shows how to create a view that includes the sysName (1.3.6.1.2.1.1.5) object:

```
config
```

```
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1.5 included
```

This example shows how to create a view that includes all the OIDs of a system group:

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
```

This example shows how to create a view that includes all the OIDs under the system group except the sysName object (1.3.6.1.2.1.1.5), which has been excluded:

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1.5 excluded
```

Verifying Configured Views

This example shows how to display information about the configured views:

```
RP/0/RP0/CPU0:router# show snmp view

  v1default 1.3.6.1 - included nonVolatile active
  SNMP_VIEW1 1.3.6.1.2.1.1 - included nonVolatile active
  SNMP_VIEW1 1.3.6.1.2.1.1.5 - excluded nonVolatile active
```

Creating Groups

If you do not explicitly specify a notify, read, or write view, the Cisco IOS XR software uses the v1 default (1.3.6.1). This example shows how to create a group that utilizes the default view:

```
RP/0/RP0/CPU0:router(config)# snmp-server group group-name v3 auth
```

The following configuration example shows how to create a group that has read access to all the OIDs in the system except the sysUpTime object (1.3.6.1.2.1.1.3), which has been excluded from the view applied to the group, but write access only to the sysName object (1.3.6.1.2.1.1.5):

```
!
  snmp-server view view_name1 1.3.6.1.2.1.1 included
  snmp-server view view_name1 1.3.6.1.2.1.1.3 excluded
  snmp-server view view_name2 1.3.6.1.2.1.1.5 included
  snmp-server group group_name1 v3 auth read view_name1 write view_name2
  !
```

Verifying Groups

This example shows how to verify the attributes of configured groups:

```
RP/0/RP0/CPU0:router# show snmp group
```

```
  groupname: group_name1                   security model:usm
  readview : view_name1                    writeview: view_name2
  notifyview: v1default
  row status: nonVolatile
```

Creating and Verifying Users

Given the following SNMPv3 view and SNMPv3 group configuration:

```
!
  snmp-server view view_name 1.3.6.1.2.1.1 included
  snmp-server group group_name v3 noauth read view_name write view-name
  !
```

This example shows how to create a noAuthNoPriv user with read and write view access to a system group:

```
config
  snmp-server user noauthuser group_name v3
```

Note The user must belong to a noauth group before a noAuthNoPriv user can be created.

Only one remote host can be assigned to the same username for SNMP version 3. If you configure the same username with different remote hosts, only the last username and remote host combination will be accepted and will be seen in the show running configuration. In the case of multiple SNMP managers, multiple unique usernames are required.

This example shows the same username case which only the last configuration will be accepted:

```
snmp-server user username  nervectrgrp remote 10.69.236.146 udp-port 162 v3 auth sha
<password> priv aes 128 <password>
snmp-server user username  nervectrgrp remote 10.214.127.2 udp-port 162 v3 auth sha <password>
 priv aes 128  <password>
snmp-server user username  nervectrgrp remote 10.69.236.147 udp-port 162 v3 auth sha
<password> priv aes 128 <password>
RP/0/RP0/CPU0:router# show run snmp-server user

  snmp-server user username nervectrgrp remote 10.69.236.147 udp-port 162 v3 auth sha
encrypted <password> priv aes 128 encrypted <password>
```

This example shows all 3 hosts for username1, username2, and username3 will be accepted.

:

```
snmp-server user username1  nervectrgrp remote 10.69.236.146 udp-port 162 v3 auth sha
<password> priv aes 128 <password>
snmp-server user username2  nervectrgrp remote 10.214.127.2 udp-port 162 v3 auth sha
<password> priv aes 128  <password>
snmp-server user username3  nervectrgrp remote 10.69.236.147 udp-port 162 v3 auth sha
<password> priv aes 128 <password>
RP/0/RP0/CPU0:router# show run snmp-server user

  snmp-server user batmanusr1 nervectrgrp remote 10.69.236.146 udp-port 162 v3 auth sha
```

```
encrypted <password> priv aes 128 encrypted <password>
  snmp-server user batmanusr2 nervectrgrp remote 10.214.127.2 udp-port 162 v3 auth sha
encrypted <password> priv aes 128 encrypted <password>
  snmp-server user batmanusr3 nervectrgrp remote 10.69.236.147 udp-port 162 v3 auth sha
encrypted <password> priv aes 128 encrypted <password>
```

This example shows how to verify the attributes that apply to the SNMP user:

```
RP/0/RP0/CPU0:router# show snmp user

  User name: noauthuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

Given the following SNMPv3 view and SNMPv3 group configuration:

```
!
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
  snmp-server group SNMP_GROUP1 v3 auth notify SNMP_VIEW1 read SNMP_VIEW1 write SNMP_VIEW1
 !
```

This example shows how to create a user with authentication (including encryption), read, and write view access to a system group:

```
config
  snmp-server user userv3authpriv SNMP_GROUP1 v3 auth md5 password123 priv aes 128 password123
```

Given the following SNMPv3 view and SNMPv3 group configuration:

```
!
  snmp-server view view_name 1.3.6.1.2.1.1 included
  snmp group group_name v3 priv read view_name write view_name
  !
```

This example shows how to create authNoPriv user with read and write view access to a system group:

```
RP/0/RP0/CPU0:router(config)# snmp-server user authuser group_name v3 auth md5 clear
auth_passwd
```

Note Because the group is configured at a security level of Auth, the user must be configured as “auth” at a minimum to access this group (“priv” users could also access this group). The authNoPriv user configured in this group, authuser, must supply an authentication password to access the view. In the example, auth_passwd is set as the authentication password string. Note that **clear** keyword is specified before the auth_passwd password string. The **clear** keyword indicates that the password string being supplied is unencrypted.

This example shows how to verify the attributes that apply to SNMP user:

```
RP/0/RP0/CPU0:router# show snmp user
```

```
  User name: authuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

Given the following SNMPv3 view and SNMPv3 group configuration:

```
!
  snmp view view_name 1.3.6.1.2.1.1 included
  snmp group group_name v3 priv read view_name write view_name
  !
```

This example shows how to create an authPriv user with read and write view access to a system group:

```
config
  snmp-server user privuser group_name v3 auth md5 clear auth_passwd priv des56 clear
priv_passwd
```

Note Because the group has a security level of Priv, the user must be configured as a “priv” user to access this group. In this example, the user, privuser, must supply both an authentication password and privacy password to access the OIDs in the view.

This example shows how to verify the attributes that apply to the SNMP user:

```
RP/0/RP0/CPU0:router# show snmp user

  User name: privuser
  Engine ID: localSnmpID
  storage-type: nonvolatile active
```

Configuring Trap Notifications: Example

The following example configures an SNMP agent to send out different types of traps. The configuration includes a v2c user, a noAuthNoPriv user, anauthNoPriv user, and an AuthPriv user.

Note The default User Datagram Protocol (UDP) port is 161. If you do not a specify a UDP port with the **udp-port** keyword and *port* argument, then the configured SNMP trap notifications are sent to port 161.

```
!
  snmp-server host 10.50.32.170 version 2c public udp-port 2345
  snmp-server host 10.50.32.170 version 3 auth userV3auth udp-port 2345
  snmp-server host 10.50.32.170 version 3 priv userV3priv udp-port 2345
  snmp-server host 10.50.32.170 version 3 noauth userV3noauth udp-port 2345
  snmp-server user userv2c groupv2c v2c
  snmp-server user userV3auth groupV3auth v3 auth md5 encrypted 140F0A13
  snmp-server user userV3priv groupV3priv v3 auth md5 encrypted 021E1C43 priv des56 encrypted
 1110001C
  snmp-server user userV3noauth groupV3noauth v3 LROwner
  snmp-server view view_name 1.3 included
  snmp-server community public RW
  snmp-server group groupv2c v2c read view_name
  snmp-server group groupV3auth v3 auth read view_name
  snmp-server group groupV3priv v3 priv read view_name
  snmp-server group groupV3noauth v3 noauth read view_name
  !
```

This example shows how to verify the configuration SNMP trap notification recipients host, the recipients of SNMP trap notifications. The output displays the following information:

- IP address of the configured notification host
- UDP port where SNMP notification messages are sent
- Type of trap configured
- Security level of the configured user
- Security model configured

```
config
  show snmp host

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3auth security model: v3 auth

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3noauth security model: v3 noauth

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userV3priv security model: v3 priv

  Notification host: 10.50.32.170 udp-port: 2345 type: trap
  user: userv2c security model: v2c
```

Setting an IP Precedence Value for SNMP Traffic: Example

The following example shows how to set the SNMP IP Precedence value to 7:

```
configure
  snmp-server ipv4 precedence 7
  exit

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: y
```

Setting an IP DSCP Value for SNMP Traffic: Example

The following example shows how to set the IP DSCP value of SNMP traffic to 45:

```
configure
```

```
snmp-server ipv4 dscp 45
exit

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: y
```

SNMP Context Mapping Configuration

Configuration of VRF Aware SNMP Context for Polling BGP Data

VRF awareness is usually done using existing, non-VRF aware MIB definitions. This means that MIB definition doesn't mention anything about VRFs. However they could be used within VRF context.

The VRF-awareness is done using SNMP contexts, where a SNMP context maps to a specific VRF.

Before you begin

- Ensure that MIB implementation is VRF-aware.
- Ensure that the implementation of all get requests support VRF context.

The following example configures VRF aware SNMP context to allow polling BGP data using BGP4-MIB.

```
snmp-server vrf <vrf_1> context <context_1>
snmp-server community <vrf_1> RW
snmp-server context <context_1>
snmp-server community-map <vrf_1> context <context_1>
snmp-server host <IP> traps version 2c <vrf_1>
```

Verification

The following configuration extracts BGP data from a peer VRF using context.

```
snmp-server vrf V1
 context V1_bgp
!
snmp-server community V1 RW
snmp-server context V1_bgp
snmp-server community-map V1 context V1_bgp
router bgp 65000
 nsr
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 neighbor 192.0.2.254
  remote-as 65001
  address-family ipv4 unicast
   route-policy ALL in
   route-policy ALL out
  !
 !
 vrf V1
  rd 111:111
  address-family ipv4 unicast
  !
  neighbor 192.0.2.255
   remote-as 65003
   address-family ipv4 unicast
   !
  !
 !
!
end
```

Configuration of OSPF processes Using SNMP Context

The following example configures data polling from two OSPF processes.

```
snmp-server community com1 RW
snmp-server community com2 RW
snmp-server context ctx1
snmp-server context ctx2
snmp-server community-map com1 context ctx1
snmp-server community-map com2 context ctx2
router ospf one
 snmp context ctx1
 area 0
  interface GigabitEthernet0/2/0/0
  !
 !
!
router ospf two
 snmp context ctx2
 area 0
  interface GigabitEthernet0/2/0/1
  !
 !
!
```

Configuration of OSPF Neighbour in VRF

The following example configures OSFP neighbours in VRF using SNMP context.

```
snmp-server vrf VRF_A
 context ctx1
!
snmp-server community com1 RW
snmp-server context ctx1
snmp-server community-map com1 context ctx1
router ospf core
 vrf VRF_A
  snmp context ctx1
 !
!
end
```

Additional References

The following sections provide references related to Implementing SNMP on Cisco IOS XR software.

Related Documents

Related Topic	Document Title
Cisco IOS XR SNMP commands	*SNMP Server Commands on* module of *System Management Command Reference for Cisco NCS 6000 Series Routers*

Related Topic	Document Title
MIB information	
Cisco IOS XR commands	
Getting started with Cisco IOS XR software	
Information about user groups and task IDs	*Configuring AAA Services on* module of *System Security Configuration Guide for Cisco NCS 6000 Series Routers*
Cisco IOS XR Quality of Service	*Modular QoSConfiguration Guide for Cisco NCS 6000 Series Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 3411	*An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*
RFC 3412	*Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*
RFC 3413	*Simple Network Management Protocol (SNMP) Applications*
RFC 3414	*User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3415	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
RFC 3416	*Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*
RFC 3417	*Transport Mappings for the Simple Network Management Protocol (SNMP)*
RFC 3418	*Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 5

Configuring Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is a media- and protocol-independent protocol that runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. Using CDP, you can view information about all the Cisco devices that are directly attached to the device.

This module describes the new and revised tasks you need to implement CDP on your Cisco IOS XR network.

For more information about CDP on the Cisco IOS XR software and complete descriptions of the CDP commands listed in this module, refer to Related Documents, on page 56. To locate documentation for other commands that might appear in the course of running a configuration task, search online in .

Table 8: Feature History for Implementing CDP on Cisco IOS XR Software

Release	Modification
Release 5.0.0	This feature was introduced.

This module contains the following topics:

Prerequisites for Implementing CDP

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Information About Implementing CDP

CDP is primarily used to obtain protocol addresses of neighboring devices and discover the platform of those devices. CDP can also be used to display information about the interfaces your router uses. CDP is media- and protocol-independent, and runs on all equipment manufactured by Cisco, including routers, bridges, access servers, and switches.

Use of SNMP with the CDP MIB allows network management applications to learn the device type and the SNMP agent address of neighboring devices and to send SNMP queries to those devices. CDP uses the CISCO-CDP-MIB.

CDP runs on all media that support Subnetwork Access Protocol (SNAP), including LAN, Frame Relay, and ATM physical media. CDP runs over the data link layer only. Therefore, two systems that support different network-layer protocols can learn about each other.

Each device configured for CDP sends periodic messages, known as *advertisements*, to a multicast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain time-to-live, or hold-time, information, which indicates the length of time a receiving device holds CDP information before discarding it. Each device also listens to the periodic CDP messages sent by others to learn about neighboring devices and determine when their interfaces to the media go up or down.

CDP Version-2 (CDPv2) is the most recent release of the protocol and provides more intelligent device tracking features. These features include a reporting mechanism that allows for more rapid error tracking, thereby reducing costly downtime. Reported error messages can be sent to the console or to a logging server, and can cover instances of unmatching native VLAN IDs (IEEE 802.1Q) on connecting ports, and unmatching port duplex states between connecting devices.

CDPv2 **show** commands can provide detailed output on VLAN Trunking Protocol (VTP) management domain and duplex modes of neighbor devices, CDP-related counters, and VLAN IDs of connecting ports.

Type-length-value fields (TLVs) are blocks of information embedded in CDP advertisements. Table 9: Type-Length-Value Definitions for CDPv2, on page 50 summarizes the TLV definitions for CDP advertisements.

Table 9: Type-Length-Value Definitions for CDPv2

TLV	Definition
Device-ID TLV	Identifies the device name in the form of a character string.
Address TLV	Contains a list of network addresses of both receiving and sending devices.
Port-ID TLV	Identifies the port on which the CDP packet is sent.
Capabilities TLV	Describes the functional capability for the device in the form of a device type; for example, a switch.
Version TLV	Contains information about the software release version on which the device is running.
Platform TLV	Describes the hardware platform name of the device, for example, Cisco 4500.
VTP Management Domain TLV	Advertises the system’s configured VTP management domain name-string. Used by network operators to verify VTP domain configuration in adjacent network nodes.
Native VLAN TLV	Indicates, per interface, the assumed VLAN for untagged packets on the interface. CDP learns the native VLAN for an interface. This feature is implemented only for interfaces that support the IEEE 802.1Q protocol.

TLV	Definition
Full/Half Duplex TLV	Indicates status (duplex configuration) of CDP broadcast interface. Used by network operators to diagnose connectivity problems between adjacent network elements.

How to Implement CDP on Cisco IOS XR Software

Enabling CDP

To enable CDP, you must first enable CDP globally on the router and then enable CDP on a per-interface basis. This task explains how to enable CDP globally on the router and then enable CDP on an interface.

SUMMARY STEPS

1. **configure**
2. **cdp**
3. **interface** *type interface-path-id*
4. **cdp**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**cdp** **Example:** `RP/0/RP0/CPU0:router(config)# cdp`	Enables CDP globally.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface pos 0/0/0/1`	Enters interface configuration mode.
Step 4	**cdp** **Example:** `RP/0/RP0/CPU0:router(config-if)# cdp`	Enables CDP on an interface.
Step 5	**commit**	

Modifying CDP Default Settings

This task explains how to modify the default version, hold-time setting, and timer settings.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **configure**
2. **cdp advertise v1**
3. **cdp holdtime** *seconds*
4. **cdp timer** *seconds*
5. **commit**
6. (Optional) **show cdp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**cdp advertise v1** **Example:** `RP/0/RP0/CPU0:router(config)# cdp advertise v1`	Configures CDP to use only version 1 (CDPv1) in communicating with neighboring devices. • By default, when CDP is enabled, the router sends CDPv2 packets. CDP also sends and receives CDPv1 packets if the device with which CDP is interacting does not process CDPv2 packets. • In this example, the router is configured to send and receive only CDPv1 packets.
Step 3	**cdp holdtime** *seconds* **Example:** `RP/0/RP0/CPU0:router(config)# cdp holdtime 30`	Specifies the amount of time that the receiving networking device will hold a CDP packet sent from the router before discarding it. • By default, when CDP is enabled, the receiving networking device holds a CDP packet for 180 seconds before discarding it. **Note** The CDP hold time must be set to a higher number of seconds than the time between CDP transmissions, which is set with the **cdp timer** command. • In this example, the value of hold-time for the *seconds* argument is set to 30.
Step 4	**cdp timer** *seconds* **Example:** `RP/0/RP0/CPU0:router(config)# cdp timer 20`	Specifies the frequency at which CDP update packets are sent. • By default, when CDP is enabled, CDP update packets are sent at a frequency of once every 60 seconds. **Note** A lower timer setting causes CDP updates to be sent more frequently.

	Command or Action	Purpose
		• In this example, CDP update packets are configured to be sent at a frequency of once every 20 seconds.
Step 5	**commit**	
Step 6	(Optional) **show cdp** **Example:** `RP/0/RP0/CPU0:router# show cdp`	Displays global CDP information. The output displays the CDP version running on the router, the hold time setting, and the timer setting.

Monitoring CDP

This task shows how to monitor CDP.

> **Note** The commands can be entered in any order.

SUMMARY STEPS

1. **show cdp entry** {***** | *entry-name*} [**protocol** | **version**]
2. **show cdp interface** [*type interface-path-id* | **location** *node-id*]
3. **show cdp neighbors** [*type interface-path-id* | **location** *node-id*] [**detail**]
4. **show cdp traffic** [**location** *node-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show cdp entry** {*****	*entry-name*} [**protocol**	**version**] **Example:** `RP/0/RSP0/CPU0:router# show cdp entry *`	Displays information about a specific neighboring device or all neighboring devices discovered using CDP.
Step 2	**show cdp interface** [*type interface-path-id*	**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show cdp interface pos 0/0/0/1`	Displays information about the interfaces on which CDP is enabled.
Step 3	**show cdp neighbors** [*type interface-path-id*	**location** *node-id*] [**detail**] **Example:** `RP/0/RSP0/CPU0:router# show cdp neighbors`	Displays detailed information about neighboring devices discovered using CDP.

	Command or Action	Purpose
Step 4	**show cdp traffic** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show cdp traffic`	Displays information about the traffic gathered between devices using CDP.

Examples

The following is sample output for the **show cdp neighbors** command:

```
RP/0/RP0/CPU0:router# show cdp neighbors

  Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                    S - Switch, H - Host, I - IGMP, r - Repeater

  Device ID        Local Intrfce   Holdtme  Capability  Platform  Port ID
  router1          Mg0/0/CPU0/0    177      T S         WS-C2924M Fa0/12
  router2          PO0/4/0/0       157      R           12008/GRP PO0/4/0/1
```

The following is sample output for the **show cdp neighbors** command. In this example, the optional *type instance* arguments are used in conjunction with the **detail** optional keyword to display detailed information about a CDP neighbor. The output includes information on both IPv4 and IPv6 addresses.

```
RP/0/RP0/CPU0:router# show cdp neighbors POS 0/4/0/0 detail

  -------------------------
  Device ID: uut-user
  SysName : uut-user
  Entry address(es):
  IPv4 address: 1.1.1.1
  IPv6 address: 1::1
  IPv6 address: 2::2
  Platform: cisco 12008/GRP, Capabilities: Router
  Interface: POS0/4/0/3
  Port ID (outgoing port): POS0/2/0/3
  Holdtime : 177 sec

  Version :
  Cisco IOS XR Software, Version 0.0.0[Default]
  Copyright (c) 2005 by cisco Systems, Inc.

  advertisement version: 2
```

The following is sample output for the **show cdp entry** command. In this example, the optional *entry* argument is used to display entry information related to a specific CDP neighbor.

```
RP/0/RP0/CPU0:router# show cdp entry router2

  advertisement version: 2

  -------------------------
  Device ID: router2
  SysName : router2
  Entry address(es):
  Platform: cisco 12008/GRP,  Capabilities: Router
  Interface: POS0/4/0/0
  Port ID (outgoing port): POS0/4/0/1
  Holdtime : 145 sec

  Version :
  Cisco IOS XR Software, Version 0.48.0[Default]
  Copyright (c) 2004 by cisco Systems, Inc.

  advertisement version: 2
```

The following is sample output for the **show cdp interface** command. In this example, CDP information related to Packet over SONET/SDH (POS) interface 0/4/0/0 is displayed.

```
RP/0/RP0/CPU0:router# show cdp interface pos 0/4/0/0

  POS0/4/0/0 is Up
    Encapsulation HDLC
    Sending CDP packets every 60 seconds
    Holdtime is 180 seconds
```

The following is sample output for the **show cdp traffic** command:

```
RP/0/RP0/CPU0:router# show cdp traffic

  CDP counters :
          Packets output: 194, Input: 99
          Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
          No memory: 0, Invalid packet: 0, Truncated: 0
          CDP version 1 advertisements output: 0, Input: 0
          CDP version 2 advertisements output: 194, Input: 99
          Unrecognize Hdr version: 0, File open failed: 0
```

The following is sample output for the **show cdp traffic** command. In this example, the optional **location** keyword and *node-id* argument are used to display information about the traffic gathered between devices using CDP from the specified node.

```
RP/0/RP0/CPU0:router# show cdp traffic location 0/4/cpu0

  CDP counters :
          Packets output: 16, Input: 13
          Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
          No memory: 0, Invalid packet: 0, Truncated: 0
          CDP version 1 advertisements output: 0, Input: 0
          CDP version 2 advertisements output: 16, Input: 13
          Unrecognize Hdr version: 0, File open failed: 0
```

Configuration Examples for Implementing CDP

Enabling CDP: Example

The following example shows how to configure CDP globally and then enable CDP on Packet over SONET/SDH (POS) interface 0/3/0/0:

```
cdp
  interface POS0/3/0/0
   cdp
```

Modifying Global CDP Settings: Example

The following example shows how to modify global CDP settings. In this example, the timer setting is set to 20 seconds, the hold-time setting is set to 30 seconds, and the version of CDP used to communicate with neighboring devices is set to CDPv1:

```
cdp timer 20
  cdp holdtime 30
  cdp advertise v1
```

The following example shows how to use the **show cdp** command to verify the CDP global settings:

```
RP/0/RP0/CPU0:router# show cdp

  Global CDP information:
          Sending CDP packets every 20 seconds
          Sending a holdtime value of 30 seconds
          Sending CDPv2 advertisements is not enabled
```

Additional References

The following sections provide references related to implementing CDP on Cisco IOS XR software.

Related Documents

Related Topic	Document Title
Cisco IOS XR CDP commands	*CDP Commands on Cisco IOS XR Software* module of *System Management Command Reference for Cisco NCS 6000 Series Routers*
Cisco IOS XR commands	
Getting started with Cisco IOS XR Software	
Information about user groups and task IDs	*Configuring AAA Services on Cisco IOS XR Software* module of *System Security Configuration Guide for Cisco NCS 6000 Series Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

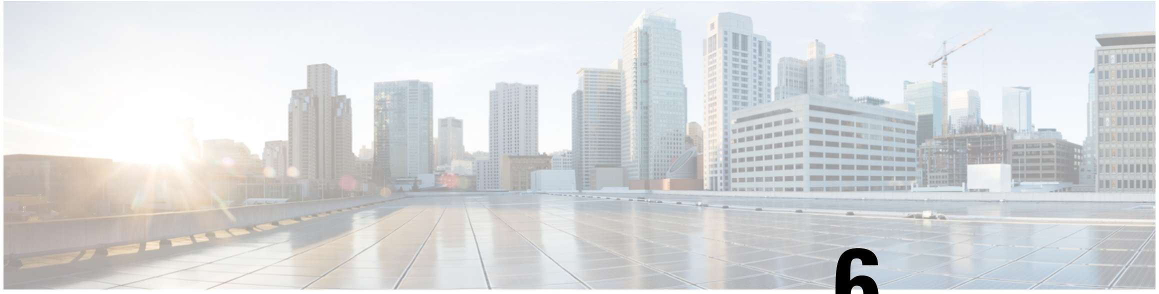
MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 6

Configuring Periodic MIB Data Collection and Transfer

This document describes how to periodically transfer selected MIB data from your router to a specified Network Management System (NMS). The periodic MIB data collection and transfer feature is also known as bulk statistics.

Table 10: Feature History for Periodic MIB Data Collection and Transfer

Release	Modification
Release 4.2.0	The periodic MIB data collection and transfer feature was introduced and supported the IF-MIB only.
Release 4.2.1	Additional MIBs were supported.

This module contains the following topics:

Prerequisites for Periodic MIB Data Collection and Transfer

To use periodic MIB data collection and transfer, you should be familiar with the Simple Network Management Protocol (SNMP) model of management information. You should also know what MIB information you want to monitor on your network devices, and the OIDs or object names for the MIB objects to be monitored.

Information About Periodic MIB Data Collection and Transfer

SNMP Objects and Instances

A type (or class) of SNMP management information is called an object. A specific instance from a type of management information is called an object instance (or SNMP variable). To configure a bulk statistics collection, you must specify the object types to be monitored using a bulk statistics object list and the specific instances of those objects to be collected using a bulk statistics schema.

MIBs, MIB tables, MIB objects, and object indices can all be specified using a series of numbers called an object identifier (OID). OIDs are used in configuring a bulk statistics collection in both the bulk statistics object lists (for general objects) and in the bulk statistics schemas (for specific object instances).

Bulk Statistics Object Lists

To group the MIB objects to be polled, you need to create one or more object lists. A bulk statistics object list is a user-specified set of MIB objects that share the same MIB index. Object lists are identified using a name that you specify. Named bulk statistics object lists allow the same configuration to be reused in different bulk statistics schemas.

All the objects in an object list must share the same MIB index. However, the objects do not need to be in the same MIB and do not need to belong to the same MIB table. For example, it is possible to group ifInOctets and a CISCO-IF-EXTENSION-MIB object in the same schema, because the containing tables for both objects are indexed by the ifIndex.

Bulk Statistics Schemas

Data selection for the Periodic MIB Data Collection and Transfer Mechanism requires the definition of a schema with the following information:

- Name of an object list.
- Instance (specific instance or series of instances defined using a wild card) that needs to be retrieved for objects in the specified object list.
- How often the specified instances need to be sampled (polling interval). The default polling interval is 5 minutes.

A bulk statistics schema is also identified using a name that you specify. This name is used when configuring the transfer options.

Bulk Statistics Transfer Options

After configuring the data to be collected, a single virtual file (VFile or *bulk statistics file*) with all collected data is created. This file can be transferred to a network management station using FTP or TFTP. You can specify how often this file should be transferred. The default transfer interval is once every 30 minutes. You can also configure a secondary destination for the file to be used if, for whatever reason, the file cannot be transferred to the primary network management station.

The value of the transfer interval is also the collection period (collection interval) for the local bulk statistics file. After the collection period ends, the bulk statistics file is frozen, and a new local bulk statistics file is created for storing data. The frozen bulk statistics file is then transferred to the specified destination.

By default, the local bulk statistics file is deleted after successful transfer to an network management station.

Benefits of Periodic MIB Data Collection and Transfer

Periodic MIB data collection and transfer (bulk statistics feature) allows many of the same functions as the bulk file MIB (CISCO-BULK-FILE-MIB.my), but offers some key advantages. The main advantage is that this feature can be configured through the CLI and does not require an external monitoring application.

Periodic MIB data collection and transfer is mainly targeted for medium to high-end platforms that have sufficient local storage (volatile or permanent) to store bulk statistics files. Locally storing bulk statistics files helps minimize loss of data during temporary network outages.

This feature also has more powerful data selection features than the bulk file MIB; it allows grouping of MIB objects from different tables into data groups (object lists). It also incorporates a more flexible instance selection mechanism, where the application is not restricted to fetching an entire MIB table.

How to Configure Periodic MIB Data Collection and Transfer

Configuring a Bulk Statistics Object List

The first step in configuring the Periodic MIB Data Collection and Transfer Mechanism is to configure one or more object lists.

SUMMARY STEPS

1. **configure**
2. **snmp-server mib bulkstat object-list** *list-name*
3. **add** {**oid** | *object-name*}
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**snmp-server mib bulkstat object-list** *list-name* **Example:** `snmp-server mib bulkstat object-list ifMib`	Defines an SNMP bulk statistics object list and enters bulk statistics object list configuration mode.

Step 3	**add** {**oid** | *object-name*}
	Example:
	`RP/0/RP0/CPU0:router(config-bulk-objects)# add`
		Adds a MIB object to the bulk statistics object list. Repeat as desired until all objects to be monitored in this list are added.

	Command or Action	Purpose

	`1.3.6.1.2.1.2.2.1.11`
	`RP/0/RP0/CPU0:router(config-bulk-objects)# add ifAdminStatus`
	`RP/0/RP0/CPU0:router(config-bulk-objects)# add ifDescr`
		Note All the objects in a bulk statistics object list have to be indexed by the same MIB index. However, the objects in the object list do not need to belong to the same MIB or MIB table.
		When specifying an object name instead of an OID (using the add command), only object names with mappings shown in the **show snmp mib object** command output can be used.

Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

What to do next

Configure a bulk statistics schema.

Configuring a Bulk Statistics Schema

The second step in configuring periodic MIB data collection and transfer is to configure one or more schemas.

Before you begin

The bulk statistics object list to be used in the schema must be defined.

SUMMARY STEPS

1. **configure**
2. **snmp-server mib bulkstat schema** *schema-name*
3. **object-list** *list-name*
4. Do one of the following:
   - **instance exact** {**interface** *interface-id* [**sub-if**] \| **oid** *oid*}
   - **instance wild** {**interface** *interface-id* [**sub-if**] \| **oid** *oid*}
   - **instance range start** *oid* **end** *oid*
   - **instance repetition** *oid* **max** *repeat-number*
5. **poll-interval** *minutes*
6. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.

Step 2	**snmp-server mib bulkstat schema** *schema-name*
	Example:
	`RP/0/RP0/CPU0:router(config)# snmp-server mib bulkstat schema intE0`
	`RP/0/RP0/CPU0:router(config-bulk-sc)#`
		Names the bulk statistics schema and enters bulk statistics schema mode.

Step 3	**object-list** *list-name* **Example:** `RP/0/RP0/CPU0:router(config-bulk-sc)# object-list ifMib`	Specifies the bulk statistics object list to be included in this schema. Specify only one object list per schema. If multiple object-list commands are executed, the earlier ones are overwritten by newer commands.

Step 4	Do one of the following:
	• **instance exact** {**interface** *interface-id* [**sub-if**] | **oid** *oid*}
	• **instance wild** {**interface** *interface-id* [**sub-if**] | **oid** *oid*}
	• **instance range start** *oid* **end** *oid*
	• **instance repetition** *oid* **max** *repeat-number*
	Example:
	`RP/0/RP0/CPU0:router(config-bulk-sc)# instance wild oid 1`
	or
	`RP/0/RP0/CPU0:router(config-bulk-sc)# instance exact interface FastEthernet 0/1.25`
	or
	`RP/0/RP0/CPU0:router(config-bulk-sc)# instance range start 1 end 2`
	or
	`RP/0/RP0/CPU0:router(config-bulk-sc)# instance repetition 1 max 4`
		Specifies the instance information for objects in this schema:
		• The **instance exact** command indicates that the specified instance, when appended to the object list, represents the complete OID.
		• The **instance wild** command indicates that all subindices of the specified OID belong to this schema. The wild keyword allows you to specify a partial, “wild carded” instance.
		• The **instance range** command indicates a range of instances on which to collect data.
		• The **instance repetition** command indicates data collection to repeat for a certain number of instances of a MIB object.
		Note Only one **instance** command can be configured per schema. If multiple **instance** commands are executed, the earlier ones are overwritten by new commands.

Step 5	**poll-interval** *minutes* **Example:** `RP/0/RP0/CPU0:router(config-bulk-sc)# poll-interval 10`	Sets how often data should be collected from the object instances specified in this schema, in minutes. The default is once every 5 minutes. The valid range is from 1 to 20000.

	Command or Action	Purpose
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

What to do next

Configure the bulk statistics transfer options.

Configuring Bulk Statistics Transfer Options

The final step in configuring periodic MIB data collection and transfer is to configure the transfer options. The collected MIB data are kept in a local file-like entity called a VFile (virtual file, referred to as a bulk statistics file in this document). This file can be transferred to a remote network management station at intervals you specify.

Before you begin

The bulk statistics object lists and bulk statistics schemas must be defined before configuring the bulk statistics transfer options.

SUMMARY STEPS

1. **configure**
2. **snmp-server mib bulkstat transfer-id** *transfer-id*
3. **buffer-size** *bytes*
4. **format** {**bulkBinary** | **bulkASCII** | **schemaASCII**}
5. **schema** *schema-name*
6. **transfer-interval** *minutes*
7. **url primary** *url*
8. **url secondary** *url*
9. **retry** *number*
10. **retain** *minutes*
11. **enable**
12. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**snmp-server mib bulkstat transfer-id** *transfer-id* **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server mib bulkstat transfer bulkstat1`	Identifies the transfer configuration with a name (*transfer-id* argument) and enters bulk statistics transfer configuration mode.
Step 3	**buffer-size** *bytes* **Example:** `RP/0/RP0/CPU0:router(config-bulk-tr)# buffersize 3072`	(Optional) Specifies the maximum size for the bulk statistics data file, in bytes. The valid range is from 1024 to 2147483647 bytes. The default buffer size is 2048 bytes. **Note** If the maximum buffer size for a bulk statistics file is reached before the transfer interval time expires, all additional data received is deleted. To correct this behavior, you can decrease the polling frequency, or increase the size of the bulk statistics buffer.

Step 4	**format** {**bulkBinary** | **bulkASCII** | **schemaASCII**}
	Example:
	`RP/0/RP0/CPU0:router(config-bulk-tr)# format schemaASCII`
		(Optional) Specifies the format of the bulk statistics data file (VFile). The default is schemaASCII.
		Note Transfers can only be performed using schemaASCII (cdcSchemaASCII) format. SchemaASCII is a human-readable format that contains parser-friendly hints for parsing data values.

Step 5	**schema** *schema-name*
	Example:
	`RP/0/RP0/CPU0:router(config-bulk-tr)# schema ATM2/0-IFMIB`
	`RP/0/RP0/CPU0:router(config-bulk-tr)# schema ATM2/0-CAR`
	`RP/0/RP0/CPU0:router(config-bulk-tr)# schema Ethernet2/1-IFMIB`
		Specifies the bulk statistics schema to be transferred. Repeat this command as desired. Multiple schemas can be associated with a single transfer configuration; all collected data are placed in a single bulk data file (VFile).

Step 6	**transfer-interval** *minutes* **Example:** RP/0/RP0/CPU0:router `RP/0/RP0/CPU0:router(config-bulk-tr)# transfer-interval 20`	(Optional) Specifies how often the bulk statistics file are transferred, in minutes. The default value is once every 30 minutes. The transfer interval is the same as the collection interval.

	Command or Action	Purpose

Step 7	**url primary** *url*
	Example:
	`RP/0/RP0/CPU0:router(config-bulk-tr)# url primary`
	`ftp://user:password@host/folder/bulkstat1`
		Specifies the network management system (host) that the bulk statistics data file is transferred to, and the protocol to use for transfer. The destination is specified as a Uniform Resource Locator (URL). FTP or TFTP can be used for the bulk statistics file transfer.

Step 8	**url secondary** *url* **Example:** `RP/0/RP0/CPU0:router(config-bulk-tr)# url secondary tftp://10.1.0.1/tftpboot/user/bulkstat1`	(Optional) Specifies a backup transfer destination and protocol for use in the event that transfer to the primary location fails. FTP or TFTP can be used for the bulk statistics file transfer.
Step 9	**retry** *number* **Example:** `RP/0/RP0/CPU0:router(config-bulk-tr)# retry 1`	(Optional) Specifies the number of transmission retries. The default value is 0 (in other words, no retries). If an attempt to send the bulk statistics file fails, the system can be configured to attempt to send the file again using this command. One retry includes an attempt first to the primary destination then, if the transmission fails, to the secondary location. For example, if the retry value is 1, an attempt is made first to the primary URL, then to the secondary URL, then to the primary URL again, then to the secondary URL again. The valid range is from 0 to 100. If all retries fail, the next normal transfer occurs after the configured transfer-interval time.
Step 10	**retain** *minutes* **Example:** `RP/0/RP0/CPU0:router(config-bulk-tr)# retain 60`	(Optional) Specifies how long the bulk statistics file should be kept in system memory, in minutes, after the completion of the collection interval and a transmission attempt is made. The default value is 0. Zero (0) indicates that the file is deleted immediately after the transfer is attempted. The valid range is from 0 to 20000. **Note** If the retry command is used, you should configure a retain interval larger than 0. The interval between retries is the retain interval divided by the retry number. For example, if **retain 10** and **retry 2** are configured, two retries are attempted once every 5 minutes. Therefore, if retain 0 is configured, no retries are attempted.
Step 11	**enable** **Example:** `RP/0/RP0/CPU0:router(config-bulk-tr)# enable`	Begins the bulk statistics data collection and transfer process for this configuration. • For successful execution of this action, at least one schema with non-zero number of objects must be configured.

	Command or Action	Purpose
		• Periodic collection and file transfer begins only if this command is configured. Conversely, the **no enable** command stops the collection process. A subsequent **enable** starts the operations again. • Each time the collection process is started using the **enable** command, data is collected into a new bulk statistics file. When the **no enable** command is used, the transfer process for any collected data immediately begins (in other words, the existing bulk statistics file is transferred to the specified management station).
Step 12	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

What to do next

Note If the maximum buffer size for a bulk statistics file is reached before the transfer interval time expires, the transfer operation is still initiated, but any bulk statistics data received after the file was full, and before it was transferred, are deleted. To correct this behavior, you can decrease the polling frequency, or increase the size of the bulk statistics buffer.

If **retain 0** is configured, no retries are attempted. This is because the interval between retries is the retain value divided by the retry value. For example, if **retain 10** and **retry 2** are configured, retries are attempted once every 5 minutes. Therefore, if you configure the retry command, you should also configure an appropriate value for the retain command.

Monitoring Periodic MIB Data Collection and Transfer

SUMMARY STEPS

1. **show snmp mib bulkstat transfer** *transfer-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show snmp mib bulkstat transfer** *transfer-name*	(Optional) The show command for this feature lists all bulk statistics virtual files (VFiles) on the system that have finished collecting data. (Data files that are not complete are not displayed.) The output lists all of the completed local bulk statistics files, the remaining time left before the bulk statistics file is deleted (remaining retention period), and the state of the bulk statistics file. The "STATE" of the bulk statistics file is one of the following: • Queued--Indicates that the data collection for this bulk statistics file is completed (in other words, the transfer interval has been met) and that the bulk statistics file is waiting for transfer to the configured destination(s). • Retry--Indicates that one or more transfer attempts have failed and that the file transfer will be attempted again. The number of retry attempts remaining are displayed in parenthesis. • Retained--Indicates that the bulk statistics file has either been successfully transmitted or that the configured number of retries have been completed. To display only the status of a named transfer (as opposed to all configured transfers), specify the name of the transfer in the transfer-name argument.

show snmp mib bulkstat transfer Sample Output

```
RP/0/RP0/CPU0:router# show snmp mib bulkstat transfer

Transfer Name : ifmib
 Retained files

 File Name                      : Time Left (in seconds)   :STATE
 ---------------------------------------------------------------------
 ifmib_Router_020421_100554683 : 173 : Retry (2 Retry attempt(s) Left)
```

Periodic MIB Data Collection and Transfer: Example

This example shows how to configure periodic MIB data collection and transfer:

```
snmp-server mib bulkstat object-list cempo
```

```
add cempMemPoolName
add cempMemPoolType
!
snmp-server mib bulkstat schema cempWild
object-list cempo
instance wild oid 8695772
poll-interval 1
!
snmp-server mib bulkstat schema cempRepeat
object-list cempo
instance repetition 8695772.1 max 4294967295
poll-interval 1
!
snmp-server mib bulkstat transfer-id cempt1
enable
url primary tftp://223.255.254.254/auto/tftp-sjc-users3/dseeniva/dumpdcm
schema cempWild
schema cempRepeat
transfer-interval 2
!
```

This example shows sample bulk statistics file content:

```
Schema-def cempt1.cempWild "%u, %s, %s, %d" Epochtime instanceoid
            1.3.6.1.4.1.9.9.221.1.1.1.1.3 1.3.6.1.4.1.9.9.221.1.1.1.1.2
cempt1.cempWild: 1339491515, 8695772.1, processor, 2
cempt1.cempWild: 1339491515, 8695772.2, reserved, 11
cempt1.cempWild: 1339491515, 8695772.3, image, 12
cempt1.cempWild: 1339491575, 8695772.1, processor, 2
cempt1.cempWild: 1339491575, 8695772.2, reserved, 11
cempt1.cempWild: 1339491575, 8695772.3, image, 12
Schema-def cempt1.cempRepeat "%u, %s, %s, %d" Epochtime instanceoid
            1.3.6.1.4.1.9.9.221.1.1.1.1.3 1.3.6.1.4.1.9.9.221.1.1.1.1.2
cempt1.cempRepeat: 1339491515, 8695772.1, processor, 2
cempt1.cempRepeat: 1339491515, 8695772.2, reserved, 11
cempt1.cempRepeat: 1339491515, 8695772.3, image, 12
cempt1.cempRepeat: 1339491515, 26932192.1, processor, 2
cempt1.cempRepeat: 1339491515, 26932192.2, reserved, 11
cempt1.cempRepeat: 1339491515, 26932192.3, image, 12
cempt1.cempRepeat: 1339491515, 35271015.1, processor, 2
cempt1.cempRepeat: 1339491515, 35271015.2, reserved, 11
cempt1.cempRepeat: 1339491515, 35271015.3, image, 12
cempt1.cempRepeat: 1339491515, 36631989.1, processor, 2
cempt1.cempRepeat: 1339491515, 36631989.2, reserved, 11
cempt1.cempRepeat: 1339491515, 36631989.3, image, 12
cempt1.cempRepeat: 1339491515, 52690955.1, processor, 2
cempt1.cempRepeat: 1339491515, 52690955.2, reserved, 11
cempt1.cempRepeat: 1339491515, 52690955.3, image, 12
```

CHAPTER 7

Configuring Network Time Protocol

Network Time Protocol (NTP) is a protocol designed to time-synchronize devices within a network. Cisco IOS XR software implements NTPv4. NTPv4 retains backwards compatibility with the older versions of NTP, including NTPv3 and NTPv2 but excluding NTPv1, which has been discontinued due to security vulnerabilities.

This module describes the tasks you need to implement NTP on the Cisco IOS XR software.

For more information about NTP on the Cisco IOS XR software and complete descriptions of the NTP commands listed in this module, see Related Documents, on page 91. To locate documentation for other commands that might appear in the course of running a configuration task, search online in .

Table 11: Feature History for Implementing NTP on Cisco IOS XR Software

Release	Modification
Release 5.0.0	This feature was introduced.

This module contains the following topics:

Prerequisites for Implementing NTP on Cisco IOS XR Software

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Information About Implementing NTP

NTP synchronizes timekeeping among a set of distributed time servers and clients. This synchronization allows events to be correlated when system logs are created and other time-specific events occur.

NTP uses the User Datagram Protocol (UDP) as its transport protocol. All NTP communication uses Coordinated Universal Time (UTC). An NTP network usually receives its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to within a millisecond of each other.

NTP uses the concept of a “stratum” to describe how many NTP “hops” away a machine is from an authoritative time source. A “stratum 1” time server typically has an authoritative time source (such as a radio or atomic clock, or a GPS time source) directly attached, a “stratum 2” time server receives its time via NTP from a “stratum 1” time server, and so on.

NTP avoids synchronizing to a machine whose time may not be accurate, in two ways. First, NTP never synchronizes to a machine that is not synchronized itself. Second, NTP compares the time reported by several machines and does not synchronize to a machine whose time is significantly different than the others, even if its stratum is lower. This strategy effectively builds a self-organizing tree of NTP servers.

The Cisco implementation of NTP does not support stratum 1 service; in other words, it is not possible to connect to a radio or atomic clock (for some specific platforms, however, you can connect a GPS time-source device). We recommend that time service for your network be derived from the public NTP servers available in the IP Internet.

If the network is isolated from the Internet, the Cisco implementation of NTP allows a machine to be configured so that it acts as though it is synchronized via NTP, when in fact it has determined the time using other means. Other machines can then synchronize to that machine via NTP.

Several manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. This software also allows UNIX-derivative servers to acquire the time directly from an atomic clock, which would subsequently propagate time information along to Cisco routers.

The communications between machines running NTP (known as *associations*) are usually statically configured; each machine is given the IP address of all machines with which it should form associations. Accurate timekeeping is made possible by exchanging NTP messages between each pair of machines with an association.

In a LAN environment, NTP can be configured to use IP broadcast messages. As compared to polling, IP broadcast messages reduce configuration complexity, because each machine can simply be configured to send or receive broadcast or multicast messages. However, the accuracy of timekeeping is marginally reduced because the information flow is one-way only.

An NTP broadcast client listens for broadcast messages sent by an NTP broadcast server at a designated IPv4 address. The client synchronizes the local clock using the first received broadcast message.

The time kept on a machine is a critical resource, so we strongly recommend that you use the security features of NTP to avoid the accidental or malicious setting of incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

When multiple sources of time (VINES, hardware clock, manual configuration) are available, NTP is always considered to be more authoritative. NTP time overrides the time set by any other method.

Note NTP associations will not be formed if the packets received are from a VRF which is different from the VRF that is configured for the NTP server or peer.

Preventing Issues due to GPS Week Number Rollover (WNRO)

- If there are no GPS sources in the NTP source chain or server chain, there is no impact of GPS Week Number Rollover (WNRO).
- GPS WNRO affects only the system clock and not user traffic.
- Contact your GPS manufacturer to fix the GPS source for this condition.

To mitigate impact of GPS sources that are subject to GPS WNRO perform the following optional workarounds:

- If the GPS source has been identified to be a cause of potential disruption on April 6, 2019 (or after), configure ntp master in the Cisco that is device connected to this source, and its clock on the Stratum 1 device to preventively isolate it. This configuration enables the device to present its own clock for synchronization to downstream NTP clients.

Note The usage of ntp master command as mentioned above is only a workaround to this condition. Use this command until the GPS source-related conditions are resolved, and to prevent the distribution of incorrect clock values throughout the network.

- Configure multiple NTP servers (ideally 4, but more than 3) at Stratum 2 level of the network, to enable NTP clients at Stratum 2 level to get clock from more than one Stratum 1 server. This way, WNRO affected Stratum 1 servers are staged to be marked as ‘false ticker’ or ‘outlier’ clock sources as compared to other non-WNRO affected Stratum 1 servers.

How to Implement NTP

Configuring Poll-Based Associations

Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

You can configure the following types of poll-based associations between the router and other devices (which may also be routers):

- Client mode
- Symmetric active mode

The client and the symmetric active modes should be used when NTP is required to provide a high level of time accuracy and reliability.

When a networking device is operating in the client mode, it polls its assigned time serving hosts for the current time. The networking device then picks a host from all the polled time servers to synchronize with. Because the relationship that is established in this case is a client-host relationship, the host does not capture or use any time information sent by the local client device. This mode is most suited for file-server and workstation clients that are not required to provide any form of time synchronization to other local clients.

Use the **server** command to individually specify the time-serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the client mode.

When a networking device is operating in the symmetric active mode, it polls its assigned time-serving hosts for the current time and it responds to polls by its hosts. Because this is a peer-to-peer relationship, the host also retains time-related information about the local networking device that it is communicating with. This mode should be used when there are several mutually redundant servers that are interconnected via diverse network paths. Most stratum 1 and stratum 2 servers on the Internet today adopt this form of network setup. Use the **peer** command to individually specify the time-serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the symmetric active mode.

When the router polls several other devices for the time, the router selects one device with which to synchronize.

Note To configure a peer-to-peer association between the router and another device, you must also configure the router as a peer on the other device.

You can configure multiple peers and servers, but you cannot configure a single IP address as both a peer and a server at the same time.

To change the configuration of a specific IP address from peer to server or from server to peer, use the **no** form of the **peer** or **server** command to remove the current configuration before you perform the new configuration. If you do not remove the old configuration before performing the new configuration, the new configuration does not overwrite the old configuration.

SUMMARY STEPS

1. **configure**
2. **ntp**
3. **server** *ip-address* [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**] [**burst**] [**iburst**]
4. **peer** *ip-address* [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**]
5. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**ntp** **Example:** `RP/0/RP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.

	Command or Action	Purpose

Step 3	**server** *ip-address* [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**] [**burst**] [**iburst**]
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# server 172.16.22.44`
	`    minpoll 8 maxpoll 12`
		Forms a server association with another system. This step can be repeated as necessary to form associations with multiple devices.

Step 4	**peer** *ip-address* [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**]
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# peer 192.168.22.33`
	`    minpoll 8 maxpoll 12 source tengige 0/0/0/1`
		Forms a peer association with another system. This step can be repeated as necessary to form associations with multiple systems.
		Note To complete the configuration of a peer-to-peer association between the router and the remote device, the router must also be configured as a peer on the remote device.

Step 5	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before`
		`  exiting(yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Broadcast-Based NTP Associates

In a broadcast-based NTP association, an NTP server propagates NTP broadcast packets throughout a network. Broadcast clients listen for the NTP broadcast packets propagated by the NTP server and do not engage in any polling.

Broadcast-based NTP associations should be used when time accuracy and reliability requirements are modest and if your network is localized and has a large number of clients (more than 20). Broadcast-based NTP associations also are recommended for use on networks that have limited bandwidth, system memory, or CPU resources. Time accuracy is marginally reduced in broadcast-based NTP associations because information flows only one way.

Use the **broadcast client** command to set your networking device to listen for NTP broadcast packets propagated through a network. For broadcast client mode to work, the broadcast server and its clients must be located on the same subnet. The time server that is transmitting NTP broadcast packets must be enabled on the interface of the given device using the **broadcast** command.

Use the **broadcast** command to set your networking device to send NTP broadcast packets.

Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

Note If you enable NTP broadcast on the physical interface, subinterface or bundle interface, then it breaks the inter-VRF Poll-Based association between client and server. As these interfaces also handle NTP unicast traffic, the interface designated as broadcast, rejects service unicast clients on it. So, NTP broadcast and NTP unicast are not allowed on the same interface.

SUMMARY STEPS

1. **configure**
2. **ntp**
3. (Optional) **broadcastdelay** *microseconds*
4. **interface** *type interface-path-id*
5. **broadcast client**
6. **broadcast** [**destination** *ip-address*] [**key** *key-id*] [**version** *number*]
7. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**ntp** **Example:** `RP/0/RP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.

	Command or Action	Purpose
Step 3	(Optional) **broadcastdelay** *microseconds* **Example:** `RP/0/RP0/CPU0:router(config-ntp)# broadcastdelay 5000`	Adjusts the estimated round-trip delay for NTP broadcasts.
Step 4	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ntp)# interface POS 0/1/0/0`	Enters NTP interface configuration mode.
Step 5	**broadcast client** **Example:** `RP/0/RP0/CPU0:router(config-ntp-int)# broadcast client`	Configures the specified interface to receive NTP broadcast packets. **Note** Go to next step to configure the interface to send NTP broadcast packets.
Step 6	**broadcast** [**destination** *ip-address*] [**key** *key-id*] [**version** *number*] **Example:** `RP/0/RP0/CPU0:router(config-ntp-int)# broadcast destination 10.50.32.149`	Configures the specified interface to send NTP broadcast packets. **Note** Go to previous step to configure the interface to receive NTP broadcast packets.

Step 7	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-ntp-int)# end`
	or
	`RP/0/RP0/CPU0:router(config-ntp-int)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring NTP Access Groups

Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

The access list-based restriction scheme allows you to grant or deny certain access privileges to an entire network, a subnet within a network, or a host within a subnet.

The access group options are scanned in the following order, from least restrictive to most restrictive:

1. **peer**—Allows time requests and NTP control queries and allows the system to synchronize itself to a system whose address passes the access list criteria.
2. **serve**—Allows time requests and NTP control queries, but does not allow the system to synchronize itself to a system whose address passes the access list criteria.
3. **serve-only**—Allows only time requests from a system whose address passes the access list criteria.
4. **query-only**—Allows only NTP control queries from a system whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first type is granted. If no access groups are specified, all access types are granted to all systems. If any access groups are specified, only the specified access types are granted.

For details on NTP control queries, see RFC 1305 (NTP version 3).

SUMMARY STEPS

1. **configure**
2. **ntp**
3. **access-group**{**peer** | **query-only** | **serve** | **serve-only**} *access-list-name*
4. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**ntp** **Example:** `RP/0/RP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**access-group**{**peer** | **query-only** | **serve** | **serve-only**} *access-list-name* **Example:**	Creates an access group and applies a basic IPv4 or IPv6 access list to it.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ntp)# access-group peer access1`	

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring NTP Authentication

This task explains how to configure NTP authentication.

Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

The encrypted NTP authentication scheme should be used when a reliable form of access control is required. Unlike the access-list-based restriction scheme that is based on IP addresses, the encrypted authentication scheme uses authentication keys and an authentication process to determine if NTP synchronization packets sent by designated peers or servers on a local network are deemed as trusted, before the time information that it carries along is accepted.

The authentication process begins from the moment an NTP packet is created. A message authentication code (MAC) is computed using the MD5 Message Digest Algorithm and the MAC is embedded into an NTP synchronization packet. The NTP synchronization packet together with the embedded MAC and key number are transmitted to the receiving client. If authentication is enabled and the key is trusted, the receiving client computes the MAC in the same way. If the computed MAC matches the embedded MAC, the system is allowed to sync to the server that uses this key in its packets.

After NTP authentication is properly configured, your networking device only synchronizes with and provides synchronization to trusted time sources.

SUMMARY STEPS

1. **configure**
2. **ntp**
3. **authenticate**
4. **authentication-key** *key-number* **md5** [**clear** | **encrypted**] *key-name*
5. **trusted-key** *key-number*
6. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**ntp** **Example:** `RP/0/RP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**authenticate** **Example:** `RP/0/RP0/CPU0:router(config-ntp)# authenticate`	Enables the NTP authentication feature.

Step 4	**authentication-key** *key-number* **md5** [**clear** | **encrypted**] *key-name*
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)#`
	`authentication-key 42`
	`md5 clear key1`
		Defines the authentication keys.
		• Each key has a key number, a type, a value, and, optionally, a name. Currently the only key type supported is **md5**.

Step 5	**trusted-key** *key-number* **Example:** `RP/0/RP0/CPU0:router(config-ntp)# trusted-key 42`	`Defines trusted authentication keys.` • If a key is trusted, this router only synchronizes to a system that uses this key in its NTP packets.
Step 6	Use one of the following commands: • **end** • **commit**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Disabling NTP Services on a Specific Interface

NTP services are disabled on all interfaces by default.

NTP is enabled globally when any NTP commands are entered. You can selectively prevent NTP packets from being received through a specific interface by turning off NTP on a given interface.

SUMMARY STEPS

1. **configure**
2. **ntp**
3. Use one of the following commands:
   - **no interface** *type interface-path-id*
   - **interface** *type interface-path-id* **disable**
4. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.

	Command or Action	Purpose
Step 2	**ntp** **Example:** `RP/0/RP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.

Step 3	Use one of the following commands:
	• **no interface** *type interface-path-id*
	• **interface** *type interface-path-id* **disable**
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# no interface pos 0/0/0/1`
	or
	`RP/0/RP0/CPU0:router(config-ntp)# interface POS 0/0/0/1 disable`
		Disables NTP services on the specified interface.

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Source IP Address for NTP Packets

By default, the source IP address of an NTP packet sent by the router is the address of the interface through which the NTP packet is sent. Use this procedure to set a different source address.

Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

SUMMARY STEPS

1. **configure**
2. **ntp**
3. **source** *type interface-path-id*
4. Use one of the following commands:
    - **end**
    - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**ntp** **Example:** `RP/0/RP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**source** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-ntp)# source POS 0/0/0/1`	Configures an interface from which the IP source address is taken. **Note** This interface is used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** keyword in the **peer** or **server** command shown in Configuring Poll-Based Associations, on page 73.

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		- Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. - Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the System as an Authoritative NTP Server

You can configure the router to act as an authoritative NTP server, even if the system is not synchronized to an outside time source.

Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

SUMMARY STEPS

1. **configure**
2. **ntp**
3. **master** *stratum*
4. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**ntp** **Example:** `RP/0/RP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**master** *stratum* **Example:**	`Makes the router an authoritative NTP server.`

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-ntp)# master 9`	**Note** Use the **master** command with caution. It is very easy to override valid time sources using this command, especially if a low stratum number is configured. Configuring multiple machines in the same network with the **master** command can cause instability in time keeping if the machines do not agree on the time.

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Updating the Hardware Clock

On devices that have hardware clocks (system calendars), you can configure the hardware clock to be periodically updated from the software clock. This is advisable for devices using NTP, because the time and date on the software clock (set using NTP) is more accurate than the hardware clock. The time setting on the hardware clock has the potential to drift slightly over time.

Note No specific command enables NTP; the first NTP configuration command that you issue enables NTP.

SUMMARY STEPS

1. **configure**
2. **ntp**

3. **update-calendar**
4. Use one of the following commands:
    - **end**
    - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.
Step 2	**ntp** **Example:** `RP/0/RP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**update-calendar** **Example:** `RP/0/RP0/CPU0:router(config-ntp)# update-calendar`	Configures the router t o update its system calendar from the software clock at periodic intervals.

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying the Status of the External Reference Clock

This task explains how to verify the status of NTP components.

Note The commands can be entered in any order.

SUMMARY STEPS

1. **show ntp associations** [**detail**] [**location** *node-id*]
2. **show ntp status** [**location** *node-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ntp associations** [**detail**] [**location** *node-id*] **Example:** `RP/0/RP0/CPU0:router# show ntp associations`	Displays the status of NTP associations.
Step 2	**show ntp status** [**location** *node-id*] **Example:** `RP/0/RP0/CPU0:router# show ntp status`	Displays the status of NTP.

Examples

The following is sample output from the **show ntp associations** command:

The following is sample output from the **show ntp status** command:

Configuration Examples for Implementing NTP

Configuring Poll-Based Associations: Example

The following example shows an NTP configuration in which the router's system clock is configured to form a peer association with the time server host at IP address 192.168.22.33, and to allow the system clock to be synchronized by time server hosts at IP address 10.0.2.1 and 172.19.69.1:

```
ntp
  server 10.0.2.1 minpoll 5 maxpoll 7
  peer 192.168.22.33

  server 172.19.69.1
```

Configuring Broadcast-Based Associations: Example

The following example shows an NTP client configuration in which interface 0/2/0/0 is configured to receive NTP broadcast packets, and the estimated round-trip delay between an NTP client and an NTP broadcast server is set to 2 microseconds:

```
ntp
  interface tengige 0/2/0/0
    broadcast client
    exit
  broadcastdelay 2
```

The following example shows an NTP server configuration where interface 0/2/0/2 is configured to be a broadcast server:

```
ntp
  interface tengige 0/2/0/2
    broadcast
```

Configuring NTP Access Groups: Example

The following example shows a NTP access group configuration where the following access group restrictions are applied:

- Peer restrictions are applied to IP addresses that pass the criteria of the access list named peer-acl.
- Serve restrictions are applied to IP addresses that pass the criteria of access list named serve-acl.
- Serve-only restrictions are applied to IP addresses that pass the criteria of the access list named serve-only-acl.
- Query-only restrictions are applied to IP addresses that pass the criteria of the access list named query-only-acl.

```
ntp
  peer 10.1.1.1
  peer 10.1.1.1
  peer 10.2.2.2
  peer 10.3.3.3
  peer 10.4.4.4
  peer 10.5.5.5
  peer 10.6.6.6
  peer 10.7.7.7
  peer 10.8.8.8
  access-group peer peer-acl
  access-group serve serve-acl
  access-group serve-only serve-only-acl
  access-group query-only query-only-acl
  exit
ipv4 access-list peer-acl
  10 permit ip host 10.1.1.1 any
  20 permit ip host 10.8.8.8 any
  exit
ipv4 access-list serve-acl
  10 permit ip host 10.4.4.4 any
  20 permit ip host 10.5.5.5 any
  exit
```

```
ipv4 access-list query-only-acl
  10 permit ip host 10.2.2.2 any
  20 permit ip host 10.3.3.3 any
  exit
ipv4 access-list serve-only-acl
  10 permit ip host 10.6.6.6 any
  20 permit ip host 10.7.7.7 any
  exit
```

Configuring NTP Authentication: Example

The following example shows an NTP authentication configuration. In this example, the following is configured:

- NTP authentication is enabled.
- Two authentication keys are configured (key 2 and key 3).
- The router is configured to allow its software clock to be synchronized with the clock of the peer (or vice versa) at IP address 10.3.32.154 using authentication key 2.
- The router is configured to allow its software clock to be synchronized with the clock by the device at IP address 10.32.154.145 using authentication key 3.
- The router is configured to synchronize only to systems providing authentication key 3 in their NTP packets.

```
ntp
  authenticate
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
  authentication-key 3 md5 encrypted 1311121E074110232621
  trusted-key 3
  server 10.3.32.154 key 3
  peer 10.32.154.145 key 2
```

Disabling NTP on an Interface: Example

The following example shows an NTP configuration in which 0/2/0/0 interface is disabled:

```
ntp
  interface tengige 0/2/0/0
    disable
    exit
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
  authentication-key 3 md5 encrypted 1311121E074110232621
  authenticate
  trusted-key 3
  server 10.3.32.154 key 3
  peer 10.32.154.145 key 2
```

Configuring the Source IP Address for NTP Packets: Example

The following example shows an NTP configuration in which Ethernet management interface 0/0/CPU0/0 is configured as the source address for NTP packets:

```
ntp
  authentication-key 2 md5 encrypted 06120A2D40031D1008124
  authentication-key 3 md5 encrypted 1311121E074110232621
  authenticate
  trusted-key 3
  server 10.3.32.154 key 3
  peer 10.32.154.145 key 2
  source MgmtEth0/0/CPU0/0
```

Configuring the System as an Authoritative NTP Server: Example

The following example shows a NTP configuration in which the router is configured to use its own NTP master clock to synchronize with peers when an external NTP source becomes unavailable:

```
ntp
  master 6
```

Updating the Hardware Clock: Example

The following example shows an NTP configuration in which the router is configured to update its hardware clock from the software clock at periodic intervals:

```
ntp
  server 10.3.32.154
  update-calendar
```

FQDN for NTP Server

NTP on Cisco IOS XR Software supports configuration of servers and peers using their Fully Qualified Domain Names (FQDN). While configuring, the FQDN is resolved via DNS into its corresponding IPv4 or IPv6 address and is stored in the running-configuration of the system. NTP supports FQDN for both IPv4 and IPv6 protocols. You can configure FQDN on default vrf.

Configure FQDN for NTP server

Configuration Example for FQDN on NTP Server on Default VRF

Use the **ntp server** command with the FQDN name to configure FQDN on default VRF. You dont need to specify VRF name. In the following example, time.cisco.com is the FQDN.

```
Router#configure
Router(config)#ntp server time.cisco.com
Router(config)#commit
```

Note When you are configuring FQDN over default VRF, you don't need to specify VRF name.

Running Configuration

Use the **show running-config ntp** command to see the ntp running configuration.

```
Router#show running-config ntp
ntp
 server 10.48.59.212
!
```

Verification

Use the **show ntp associations** command to verify that an NTP association has come up.

```
Router#show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
 ~10.48.59.212     173.38.201.67     2    42   128    3  196.06  -14.25  3949.4
 * sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

Additional References

The following sections provide references related to implementing NTP on Cisco IOS XR software.

Related Documents

Related Topic	Document Title
Cisco IOS XR clock commands	*Clock Commands on* module of *System Management Command Reference for Cisco NCS 6000 Series Routers*
Cisco IOS XR NTP commands	*NTP Commands on* module of *System Management Command Reference for Cisco NCS 6000 Series Routers*
Information about getting started with Cisco IOS XR Software	
Cisco IOS XR master command index	
Information about user groups and task IDs	*Configuring AAA Services on* module of *System Security Configuration Guide for Cisco NCS 6000 Series Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 1059	*Network Time Protocol, Version 1: Specification and Implementation*
RFC 1119	*Network Time Protocol, Version 2: Specification and Implementation*
RFC 1305	*Network Time Protocol, Version 3: Specification, Implementation, and Analysis*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 8

Configuring Network Configuration Protocol

This module provides details of the Network Configuration Protocol. For relevant commands, see *System Security Command Reference for Cisco NCS 6000 Series Routers*.

Release	Modification
Release 5.3.0	This feature was introduced.
Release 5.3.1	Support extended for more Yang models.
Release 6.0	Support extended for the Netconf subsystem configuration to be vrf aware. The configuration of the netconf port is no longer sufficient to start the Netconf subsystem support. At least one vrf needs to be configured. The configuration of the port is now optional.

The Network Configuration Protocol

The Network Configuration Protocol (Netconf) provides mechanisms to install, manipulate, and delete the configuration of network devices. It uses an Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages. Yang is a data modeling language used with Netconf.

Netconf uses a simple RPC-based (Remote Procedure Call) mechanism to facilitate communication between a client and a server. The client can be a script or application typically running as part of a network manager. The server is typically a network device.

The configuration of features need not be done the traditional way (using CLIs), the client application (controller) reads the Yang model and communicates with the Netconf server (IOS XR) accordingly.

Note Following are the deviations from IETF-NACM YANG, where the system does not support:

- The *ordered-by-user* functionality for rule-lists and rules. rule-lists & rules are sorted based on name.
- The *enable-nacm* leaf.
- The *notification* related leafs (notification-name & denied-notifications.)

Netconf Sessions and Operations

A Netconf session is the logical connection between a network configuration application and a network device. A device should be capable of supporting multiple sessions and atleast one Netconf session.

Characteristics of a netconf session:

- Netconf is connection-oriented - SSH is the underlying transport.
- The netconf client establishes session with the server.
- Netconf sessions are established with the *hello* message. Features and capabilities are announced.
- Sessions can be terminated using the *close* or *kill* messages.

Basic Netconf operations:

- Get configuration <get-config>
- Get all information <get>
- Edit configuration <edit-config>
- Copy configuration <copy-config>

Note <copy-config> does not support source attribute with “data store” at present.

- <lock>, <unlock>
- <kill-session>
- <close-session>
- Commit configuration <commit>

The Yang data model

Each feature has a defined Yang Model which is synthesized from the schemas. A model is published in a tree format and includes:

- Top level nodes and their subtrees
- Subtrees that augment nodes in other yang models

```
Example: The aaa Yang model
module: Cisco-IOS-XR-aaa-lib-cfg
   +--rw aaa
      +--rw accountings
      |  +--rw accounting* [type listname]
      |     +--rw type                 xr:Cisco-ios-xr-string
      |     +--rw listname             xr:Cisco-ios-xr-string
      |     +--rw rp-failover?         Aaa-accounting-rp-failover
      |     +--rw broadcast?           Aaa-accounting-broadcast
      |     +--rw type-xr?             Aaa-accounting
      |     +--rw method*              Aaa-method
      |     +--rw server-group-name*   string
      +--rw authorizations
      |  +--rw authorization* [type listname]
      |     +--rw type                 xr:Cisco-ios-xr-string
      |     +--rw listname             xr:Cisco-ios-xr-string
      |     +--rw method*              Aaa-method
      |     +--rw server-group-name*   string
      +--rw accounting-update!
      |  +--rw type                 Aaa-accounting-update
      |  +--rw periodic-interval?   uint32
      +--rw authentications
         +--rw authentication* [type listname]
            +--rw type                 xr:Cisco-ios-xr-string
            +--rw listname             xr:Cisco-ios-xr-string
            +--rw method*              Aaa-method
            +--rw server-group-name*   string
```

Advantages of using the Yang model are:

- Yang supports programmatic interfaces.
- Yang supports simplified network management applications.
- Yang supports interoperability that provides a standard way to model management data.

Netconf and Yang

The workflow displayed here, will help the user to understand how Netconf-Yang can configure and control the network with minimal user intervention. The required components:

- Cisco Router (ASR9000 series or CRS) with Netconf capability
- Netconf Client Application with connection to the router

S. No.	Device / component	Action
1	Cisco router (ASR 9000 or CRS router)	Login/ access the router.
2	Cisco router	Prerequisites for enabling Netconf. • k9sec pie must be installed. • Crypto keys must be generated.

S. No.	Device / component	Action
3	Cisco router	Enable Netconf agent. Use the **netconf-yang agent ssh** and **ssh server netconf** command. The port can be selected. By default, it is set as 830.
4	Cisco router	Yang models are a part of the software image. The models can be retrieved from the router , using the <get-schema> operation.
5	Netconf client (application) The application can be on any standalone application or a SDN controller supporting Netconf	Installs and processes the Yang models. The client can offer a list of supported yang models; else the user will have to browse and locate the required yang file. There is a yang model file for each configuration module; for instance if the user wants to configure CDP , the relevant yang model is Cisco-IOS-XR-cdp-cfg **Note** Refer the table which lists all the supported yang models. Supported Yang Models , on page 96
5	Netconf client	Sends Netconf operation request over SSH to the router. A configuration request could include Yang-based XML data to the router. Currently, SSH is the only supported transport method.
6	Cisco router	Understands the Yang-based XML data and the network is configured accordingly (in case of configuration request from the client).
		The interactions between the client and the router happens until the network is configured as desired.

Supported Yang Models

The Yang models can be downloaded from a prescribed location (ftp server) or can also be retrieved directly from the router using the get-schema operation.

For a feature, separate Yang models are available for configuring the feature and to get operational statistics (show commands). The **-cfg.yang** suffix denotes configuration and **-oper*.yang** is for operational data statistics. In some cases, **-oper** is followed by **-sub**, indicating that a submodule(s) is available.

For a list of supported Yang models, see https://github.com/YangModels/yang/tree/master/vendor/cisco/xr

Denial of Services Defence for Netconf-Yang

In case of a DoS (Denial of Service) attack on Netconf, wherein, Netconf receives numerous requests in a short span of time, the router may become irresponsive if Netconf consumes most of the bandwidth or CPU processing time. This can be prevented, by limiting the traffic directed at the Netconf agent. This is achieved using the **netconf-yang agent rate-limit** and **netconf-yang agent session** commands.

If rate-limit is set, the Netconf processor measures the incoming traffic from the SSH server. If the incoming traffic exceeds the set rate-limit, the packets are dropped.

If session-limit is set, the Netconf processor checks for the number of open sessions. If the number of current sessions is greater than or equal to, the set limit, no new sessions are opened.

Session idle- timeout and absolute-timeout also prevent DoS attacks. The Netconf processor closes the sessions, even without user input or intervention, as soon at the time out session is greater than or equal to the set time limit.

The relevant commands are discussed in detail, in the *System Security Command Reference for Cisco NCS 6000 Series Routers*

Enabling NETCONF over SSH

This task enables NETCONF over SSH. SSH is currently the only supported transport method .

If the client supports, Netconf over ssh can utilize the multi-channeling capabilities of IOS XR ssh server. For additional details about Multi-channeling in SSH, see *Implementing Secure Shell* in *System Security Configuration Guide*.

Prerequisites:

- k9sec pie must be installed, otherwise the port configuration for the netconf ssh server cannot be completed. (The Netconf subsystem for SSH, as well as, SSH cannot be configured without the k9sec pie.)
- Crypto keys must be generated prior to this configuration.
- The Netconf-YANG feature is packaged in the mgbl pie, which must be installed before enabling the Netconf-YANG agent.

SUMMARY STEPS

1. **configure**
2. **netconf-yang agent ssh**
3. **ssh server netconf** [**vrf** *vrf-name* [**ipv4 access-list***ipv4 access list name*] [**ipv6 access-list** *ipv6 access list name*]]
4. **ssh server netconf port** *port-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters XR Config mode.

	Command or Action	Purpose
Step 2	**netconf-yang agent ssh** **Example:** `RP/0/RP0/CPU0:router (config) # netconf agent ssh`	Enables NETCONF agent over SSH connection. After NETCONF is enabled, the Yang model in the controllcker, can configure the relevant models. **Note** The Yang models can be retrieved from the router via NETCONF <get-schema> operation.
Step 3	**ssh server netconf** [**vrf** *vrf-name* [**ipv4 access-list***ipv4 access list name*] [**ipv6 access-list** *ipv6 access list name*]] **Example:** `RP/0/RP0/CPU0:router (config) # ssh server netconf vrf netconfvrf ipv4 access-list InternetFilter`	Brings up the netconf subsytem support with SSH server using a specified VRF of up to 32 characters. If no VRF is specified, the default VRF is used. To stop the SSH server from receiving any further connections for the specified VRF, use the **no** form of this command. Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the netconf subsystem of the ssh server before the port is opened. **Note** The netconf subsystem support with SSH server can be configured for use with multiple VRFs .
Step 4	**ssh server netconf port** *port-number* **Example:** `RP/0/RP0/CPU0:router (config) # ssh server netconf port 830`	Configures a port for the netconf ssh server. This command is optional. If no port is specified, port 830 is uses by default. **Note** 830 is the IANA-assigned TCP port for NETCONF over SSH, but it can be changed using this command.

What to do next

The **show netconf-yang statistics** command and **show netconf-yang clients** command can be used to verify the configuration details of the netconf agent.

The **clear netconf-yang agent session** command clears the specified Netconf session (on the Netconf server side).

Examples: Netconf over SSH

This section illustrates some examples relevant to Netconf:

Enabling netconf-yang for ssh transport and netconf subsystem for default vrf with default port (830)

```
config
netconf-yang agent ssh
ssh server netconf vrf default
  !
!
```

Enabling netconf-yang for ssh transport and netconf subsystem for vrf *green* and vrf *red* with netconf port (831)

```
config
netconf-yang agent ssh
!
ssh server netconf vrf green
ssh server netconf vrf red
ssh server netconf port 831
  !
!
```

Show command outputs

```
show netconf-yang statistics
Summary statistics        requests|             total time|   min time per request|   max
time per request|   avg time per request|
other                             0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
close-session                     4|       0h  0m  0s   3ms|       0h  0m  0s   0ms|
 0h  0m  0s   1ms|       0h  0m  0s   0ms|
kill-session                      0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
get-schema                        0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
get                               0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s
get-config                        1|       0h  0m  0s   1ms|       0h  0m  0s   1ms|
 0h  0m  0s   1ms|       0h  0m  0s   1ms|
edit-config                       3|       0h  0m  0s   2ms|       0h  0m  0s   0ms|
 0h  0m  0s   1ms|       0h  0m  0s   0ms|
commit                            0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
cancel-commit                     0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
lock                              0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
unlock                            0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
discard-changes                   0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
validate                          0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
```

```
show netconf-yang clients
client session ID|   NC version|    client connect time|        last OP time|        last
OP type|    <lock>|
22969|                 1.1|         0d  0h  0m  2s|             11:11:24|
close-session|        No|
15389|                1.1|        0d  0h  0m  1s|            11:11:25|         get-config|
           No|
```

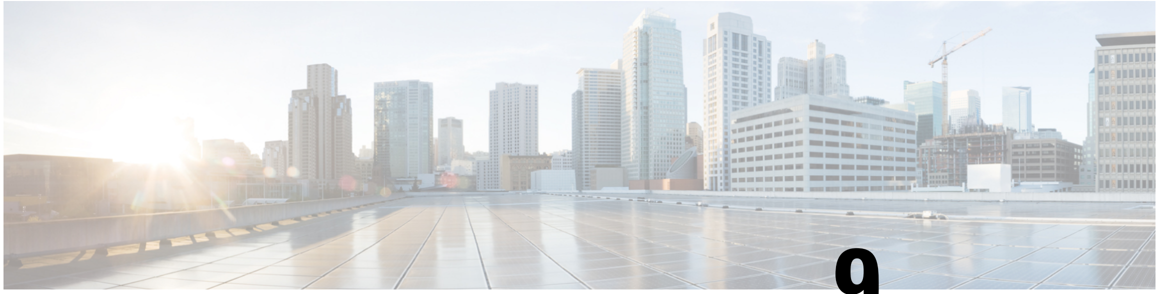
Additional Reference

Table 12: Related Documents

Related Topic	Document Title
Netconf-Yang	For related commands, see *System Security Command Reference for Cisco NCS 6000 Series Routers*

Table 13: Standards

Component	RFCs
YANG	6020
NETCONF	6241
NETCONF over SSH	6242



CHAPTER 9

Configuring Secure Domain Routers

Secure Domain Routers (SDRs) are a means of dividing a single physical system into multiple logically separated routers.

Table 14: Feature History for Configuring Multiple Secure Domain Routers

Release	Modification
Release 5.0.0	SDR feature was introduced.
Release 6.1.2	Support was added for multi-SDRs on single-chassis system.
Release 6.3.1	Support was added for multi-SDRs on multi-chassis system.

This module contains the following topics:

What Is a Secure Domain Router?

Cisco routers running the Cisco IOS XR software can be partitioned into multiple independent routers known as Secure Domain Routers (SDRs). An user defined SDR is termed as named-SDR.

SDRs are a means of dividing a single physical system into multiple logically separated routers. The SDRs are spawned as Virtual Machines (VMs). Each SDR performs routing functions similar to a physical router, but they share resources with the rest of the system. For example, the software image, configurations, protocols, and routing tables are unique to a particular SDR. Other system functions, including chassis-control and switch fabric, are shared with the rest of the system.

On Cisco NCS 6000 Series routers, multiple SDRs (multi-SDR) can be created. A maximum of three SDRs are supported. A part of system resource like line cards, memory, CPUs are allocated to each SDR. By creating multiple SDRs, the system is converted from Single Owner Single Tenant (SOST) to Single Owner Multiple

Tenant (SOMT) unit. Each SDR operates as independent unit. Hence, they are administered and managed individually. Also individual SDRs can be independently upgraded or downgraded as per need.

For more information on SDR attributes, see Multi-SDR Environment, on page 112.

For more information on SDR software upgrade, see Software Upgrade in Multi-SDR Environment, on page 113.

Create Multiple Secure Domain Routers

Creation of multiple named-SDRs involves these three stages:

1. Delete the default-SDR
2. Create a named-SDR
3. Assign inventory to the named-SDR

Note A maximum of three named-SDRs can be created.

Multi-SDR Prerequisites

Before configuring multiple Secure Domain Routers (SDRs), the following conditions must be met:

Software Version Requirements

- Multi-SDRs are supported only on NCS-6008 single-chassis running Cisco IOS XR, Release 6.1.2 and later.
- Multi-SDRs are supported only on NCS-6008 multi-chassis running Cisco IOS XR, Release 6.3.1 and later.

Required Cards for each SDR

A set of operational Line Cards (LC) and Route Processor (RP).

Initial Setup

- Uninstall inactive packages and SMUs from XR VM and System Admin VM.
- Install System Admin VM mandatory SMU (if any).
- Verify all the nodes are in operational state by using the **show platform** command and ensure basic system stability.
- Take back-up of the contents in the hard disk before converting the system into multi-sdr. The contents will be lost during the hard disk partition among the named-SDRs.
- Ensure connectivity to all the three console ports on RP faceplate. For more information on console ports, see Console Access to Named-SDRs , on page 109.

Delete Default-SDR

By default, the system will start up with a single default-SDR, which is an SOST environment. To configure named-SDRs, the default-SDR must be deleted, which enables the system to convert from an SOST to an SOMT environment.

Step 1 **config**

Example:

```
sysadmin-vm:0_RP0# config
```

Enters XR Config mode.

Step 2 **no sdr default-sdr**

Example:

```
sysadmin-vm:0_RP0(config)# no sdr default-sdr
```

Removes the default-SDR from the system.

Step 3 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Example: Delete default-SDR

```
sysadmin-vm:0_RP0# config
Thu Aug  11 00:22:16.580 UTC
Entering configuration mode terminal
sysadmin-vm:0_RP0(config)# no sdr default-sdr
Thu Aug  11 00:22:20.864 UTC
sysadmin-vm:0_RP0(config)# commit
Thu Aug  11 00:22:24.595 UTC
Commit complete.
```

What to do next

Verify that the default-SDR is deleted. Run the **show sdr** command. If the default SDR is deleted, no SDR entries will be found.

```
sysadmin-vm:0_RP0# show running-config sdr
Sun Dec  13 13:17:20.530 UTC
% No entries found.

sysadmin-vm:0_RP0# show sdr
Sun Dec  13 13:17:22.750 UTC
```

```
%  No entries found.
```

Configure Multiple Secure Domain Routers

In this task you will configure a named-SDR and allocate inventory. The inventory includes RP resources (memory and CPU) and line cards. You can repeat the steps to create maximum of three named-SDRs.

Note

- The SDR boundary is defined at the line card level. Hence, it is necessary to allocate at least one line card to each SDR. A single line card cannot be shared between multiple SDRs. Fabric cards are shared implicitly among named-SDRs.
- If you configure an SDR with IPv4 and IPv6 ACL scale configurations and if we reload the same ACL scale configurations without clearing the previous ACL scale configurations, then the limits are breached and the configuration fails to load onto the SDR.

Note When creating multiple SDRs, the install commit may not work as expected sometimes because of resource constraints.

Before you begin

Before you configure named-SDRs, the default-SDR must be deleted.

Step 1 **config**

Example:

```
sysadmin-vm:0_RP0# config
```

Enters system administration configuration mode.

Step 2 **sdr** *sdr-name*

Example:

```
sysadmin-vm:0_RP0(config)#  sdr VRFPE-SDR1
```

Creates a named-SDR and Enters SDR configuration mode.

In the following steps, you will add resources to the named-SDR.

Step 3 **resources card-type RP**

Example:

```
sysadmin-vm:0_RP0(config-sdr-VRFPE-SDR1)#  resources card-type RP
```

Enters RP resources allocation mode.

Step 4 **vm-memory** *unit*

Example:

```
sysadmin-vm:0_RP0(config-card-type-RP)#  vm-memory 11
```

Allocates RP memory to the named-SDR. Unit of VM memory size is in GB. Recommended memory value for each named-SDR = 11.

Step 5 **vm-cpu** *number-of-CPUs*

Example:

```
sysadmin-vm:0_RP0(config-card-type-RP)#  vm-cpu 4
```

Allocates RP CPUs to the named-SDR.

Number of CPUs:

- Default value for each named-SDR = 4
- Configurable minimum value = 2
- Configurable maximum value = 6

We recommend that you use the default CPU value. Change the default value only when necessary.

Step 6 **location** *node-id*

Example:

```
sysadmin-vm:0_RP0(config-sdr-VRFPE-SDR1)#  location 0/RP0
```

Allocates first RP to the named-SDR based on the specified RP location.

Step 7 **location** *node-id*

Example:

```
sysadmin-vm:0_RP0(config-location-0/RP0)#  location 0/RP1
```

Allocates second RP to the named-SDR to be used for redundancy.

Step 8 **exit**

Example:

```
sysadmin-vm:0_RP0(config-location-0/RP1)#  exit
```

Exits the RP configuration mode and returns to named-SDR configuration mode.

Step 9 **location** *node-id*

Example:

```
sysadmin-vm:0_RP0(config-sdr-VRFPE-SDR1)#  location 0/0
```

Allocates line card to the named-SDR based on the specified LC location.

Note The same LC cannot be allocated to multiple SDRs.

Step 10 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.

- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Example: Named-SDR

Creating a single named-SDR.

```
sysadmin-vm:0_RP0# config
sysadmin-vm:0_RP0(config)# sdr VRFPE-SDR1
sysadmin-vm:0_RP0(config-sdr-VRFPE-SDR1)#  resources card-type RP vm-memory 11 vm-cpu 4
sysadmin-vm:0_RP0(config-sdr-VRFPE-SDR1)#  location 0/RP0
sysadmin-vm:0_RP0(config-location-0/RP0)#  location 0/RP1
sysadmin-vm:0_RP0(config-location-0/RP1)#  exit
sysadmin-vm:0_RP0(config-sdr-VRFPE-SDR1)#  location 0/0
sysadmin-vm:0_RP0(config-sdr-VRFPE-SDR1)# commit
```

Creating three named-SDRs.

```
sysadmin-vm:0_RP0# config
Thu Aug  11 00:22:16.580 UTC
Entering configuration mode terminal
sysadmin-vm:0_RP0(config)# sdr VRFPE-SDR1
sysadmin-vm:0_RP0(config-sdr-VRFPE-SDR1)#  resources card-type RP
sysadmin-vm:0_RP0(config-card-type-RP)#   vm-memory 11
sysadmin-vm:0_RP0(config-card-type-RP)#   vm-cpu    4
sysadmin-vm:0_RP0(config-card-type-RP)# location 0/RP0
sysadmin-vm:0_RP0(config-location-0/RP0)#  location 0/RP1
sysadmin-vm:0_RP0(config-location-0/RP1)# location 0/0

sysadmin-vm:0_RP0(config-location-0/0)# sdr Internet-SDR
sysadmin-vm:0_RP0(config-sdr-Internet-SDR)#  resources card-type RP
sysadmin-vm:0_RP0(config-card-type-RP)#   vm-memory 11
sysadmin-vm:0_RP0(config-card-type-RP)#   vm-cpu    4
sysadmin-vm:0_RP0(config-card-type-RP)# location 0/RP0
sysadmin-vm:0_RP0(config-location-0/RP0)# location 0/RP1
sysadmin-vm:0_RP0(config-location-0/RP1)# location 0/6

sysadmin-vm:0_RP0(config-location-0/6)# sdr P-SDR
sysadmin-vm:0_RP0(config-sdr-P-SDR)#  resources card-type RP
sysadmin-vm:0_RP0(config-card-type-RP)#   vm-memory 11
sysadmin-vm:0_RP0(config-card-type-RP)#   vm-cpu    4
sysadmin-vm:0_RP0(config-card-type-RP)# location 0/RP0
sysadmin-vm:0_RP0(config-location-0/RP0)#  location 0/RP1
sysadmin-vm:0_RP0(config-location-0/RP1)# location 0/1

sysadmin-vm:0_RP0(config-location-0/1)# commit
Thu Aug  11 00:31:20.827 UTC
Commit complete.

sysadmin-vm:0_RP0(config-location-0/1)#
System message at 2016-08-11 00:31:21...
Commit performed by admin via tcp using system.
sysadmin-vm:0_RP0(config-location-0/1)# end
Thu Aug  11 00:31:23.455 UTC

sysadmin-vm:0_RP0# 0/6/ADMIN0:Aug 11 00:32:48.488 : vm_manager[2907]: %INFRA-VM_MANAGE Info:
 vm_manager started VM Internet-SDR--1
```

```
0/0/ADMIN0:Aug 11 00:32:48.810 : vm_manager[2902]: %INFRA-VM_MANAGER-4-INFO : Info: vmtarted
 VM VRFPE-SDR1--1
0/RP1/ADMIN0:Aug 11 00:33:01.075 : vm_manager[3162]: %INFRA-VM_MANAGER-4-INFO : Info:
started VM Internet-SDR--1
0/RP0/ADMIN0:Aug 11 00:33:12.019 : vm_manager[3183]: %INFRA-VM_MANAGER-4-INFO : Info:
started VM Internet-SDR--1
0/1/ADMIN0:Aug 11 00:33:19.744 : vm_manager[2917]: %INFRA-VM_MANAGER-4-INFO : Info: vmtarted
 VM P-SDR--1
0/RP1/ADMIN0:Aug 11 00:34:38.562 : vm_manager[3162]: %INFRA-VM_MANAGER-4-INFO : Info:
started VM P-SDR--2
0/RP0/ADMIN0:Aug 11 00:35:00.487 : vm_manager[3183]: %INFRA-VM_MANAGER-4-INFO : Info:
started VM P-SDR--2
0/RP1/ADMIN0:Aug 11 00:36:18.683 : vm_manager[3162]: %INFRA-VM_MANAGER-4-INFO : Info:
started VM VRFPE-SDR1--3
0/RP0/ADMIN0:Aug 11 00:36:54.481 : vm_manager[3183]: %INFRA-VM_MANAGER-4-INFO : Info:
started VM VRFPE-SDR1--3
```

Running configuration for configuring three named-SDRs:

```
sysadmin-vm:0_RP0# show run sdr
Tue Aug  16 18:50:51.835 UTC
sdr Internet-SDR
 resources card-type RP
  vm-memory 11
  vm-cpu    4
 !
 location 0/6
 !
 location 0/RP0
 !
 location 0/RP1
 !
!
sdr P-SDR
 resources card-type RP
  vm-memory 11
  vm-cpu    4
 !
 location 0/1
 !
 location 0/RP0
 !
 location 0/RP1
 !
!
sdr VRFPE-SDR1

 resources card-type RP
  vm-memory 11
  vm-cpu    4
 !
 location 0/0
 !
 location 0/RP0
 !
 location 0/RP1
 !
!
```

What to do next

After the named-SDR are created, verify the VM state for each SDR.

Execute the **show sdr** command to check that the Status is "RUNNING" for all VMs in each SDR.

```
sysadmin-vm:0_RP0# show sdr

Wed Aug  17 16:01:06.626 UTC

SDR: Internet-SDR
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
0/RP0/VM1    192.0.0.4       RUNNING          1           08/11/2016 00:33:12
0/RP1/VM1    192.0.4.4       RUNNING          1           08/11/2016 00:33:01
0/6/VM1      192.0.88.3      RUNNING          1           08/11/2016 00:32:48

SDR: P-SDR
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
0/RP0/VM2    192.0.0.6       RUNNING          2           08/11/2016 03:24:43
0/RP1/VM2    192.0.4.6       RUNNING          2           08/11/2016 03:24:32
0/1/VM1      192.0.68.3      RUNNING          2           08/11/2016 03:25:26

SDR: VRFPE-SDR1
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
0/RP0/VM3    192.0.0.8       RUNNING          2           08/11/2016 02:32:15
0/RP1/VM3    192.0.4.8       RUNNING          2           08/11/2016 02:32:23
0/0/VM1      192.0.64.3      RUNNING          2           08/11/2016 02:32:40
```

Execute the **show vm** command to check that the Status for named-SDRs is "running" at the line card and RP locations.

```
sysadmin-vm:0_RP0# show vm

Wed Aug  17 16:01:20.239 UTC

Location: 0/0
Id                Status        IP Address       HB Sent/Recv
-------------------------------------------------------------
sysadmin          running       192.0.64.1       NA/NA
VRFPE-SDR1        running       192.0.64.3       58375/58375

Location: 0/1
Id                Status        IP Address       HB Sent/Recv
-------------------------------------------------------------
sysadmin          running       192.0.68.1       NA/NA
P-SDR             running       192.0.68.3       58360/58360

Location: 0/6
Id                Status        IP Address       HB Sent/Recv
-------------------------------------------------------------
sysadmin          running       192.0.88.1       NA/NA
Internet-SDR      running       192.0.88.3       58401/58401

Location: 0/RP0
Id                Status        IP Address       HB Sent/Recv
-------------------------------------------------------------
sysadmin          running       192.0.0.1        NA/NA
Internet-SDR      running       192.0.0.4        1169260/1169260
P-SDR             running       192.0.0.6        1146966/1146966
VRFPE-SDR1        running       192.0.0.8        1146729/1146729
```

```
Location: 0/RP1
Id                Status        IP Address       HB Sent/Recv
-------------------------------------------------------------
sysadmin          running       192.0.4.1        NA/NA
Internet-SDR      running       192.0.4.4        1167934/1167934
P-SDR             running       192.0.4.6        1147031/1147031
VRFPE-SDR1        running       192.0.4.8        1146789/1146789
```

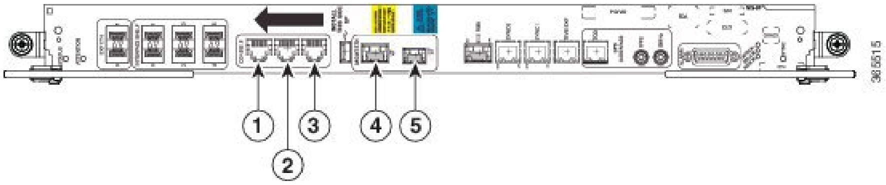
To view details about a specific SDR, use the **show sdr <sdr-name> detail** command in System Admin EXEC mode.

Console Access to Named-SDRs

By default, console1 on active RP is used to access the XR VM. With named-SDRs, you can either use console1 or console2 of an active RP to access any of the named-SDR (XR VM). You can connect two named-SDRs (XRs) at any given time. The console 0 is reserved to access the System Admin VM.

Note The RP on which XR VM gets created first, becomes active RP. It can be either RP0 or RP1.

Figure 4: Faceplate of RP



Item No.	Details
1	Console0: SysAdmin Console
2	Console1: XR console for both default-SDR and named-SDR
3	Console2: XR console for named-SDR only
4	SysAdmin MGMT Ethernet port
5	XR VM MGMT Ethernet port

Setup Console Access for Named-SDR

In this task you will configure the console port to access a named-SDR.

Note You can connect to all three SDRs by applying **console-attach** command to both RPs. But, you cannot connect to all SDRs by completing the vty configuration on just one RP.

Step 1 **config**

Example:

```
sysadmin-vm:0_RP0# config
```

Enters system administration configuration mode.

Step 2 **console attach-sdr location** *node-id* **tty-name** *tty-name* **sdr-name** *sdr-name*

Example:

```
sysadmin-vm:0_RP0(config)#  console attach-sdr location 0/RP0 tty-name console1 sdr-name VRFPE-SDR1
```

Specifies the location, tty name and named-SDR that is accessed through console of active RP.

Variables:

- *node-id* specifies the location of active RP.
- *tty-name* can either be console1 or console2.
- *sdr-name* refers to the named-SDR.

Step 3 Use the **commit** or **end** command.

commit —Saves the configuration changes and remains within the configuration session.

end —Prompts user to take one of these actions:

- **Yes** — Saves configuration changes and exits the configuration session.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Example: Console Access

The following example shows the console details:

```
sysadmin-vm:0_RP0# config
sysadmin-vm:0_RP0(config)# console attach-sdr location 0/RP0 tty-name console1 sdr-name
VRFPE-SDR1
sysadmin-vm:0_RP0(config)# console attach-sdr location 0/RP1 tty-name console1 sdr-name
VRFPE-SDR1
sysadmin-vm:0_RP0(config)# console attach-sdr location 0/RP0 tty-name console2 sdr-name
P-SDR
sysadmin-vm:0_RP0(config)# console attach-sdr location 0/RP1 tty-name console2 sdr-name
P-SDR
sysadmin-vm:0_RP0(config)# commit
```

What to do next

1. Verify that each named-SDR is attached with the consoles on the RPs.

```
sysadmin-vm:0_RP0# show run console
console attach-sdr location 0/RP0
 tty-name console1
```

```
    sdr-name VRFPE-SDR1
  !
  console attach-sdr location 0/RP1
   tty-name console1
    sdr-name VRFPE-SDR1
  !
  console attach-sdr location 0/RP0
   tty-name console2
    sdr-name P-SDR
  !
  console attach-sdr location 0/RP1
   tty-name console2
    sdr-name P-SDR
  !
```

2. Telnet to each console port and check for a successful connectivity.

Console Access Mapping

This table provides a sample console access mapping matrix for three configured named-SDRs, namely, sdr1, sdr2, and sdr3.

Command	Console0 Access	Console1 Access to XR VM	Console2 Access to XR VM
On system boot-up	Access sysadmin	defaults-SDR	Unused
no sdr default-sdr (No SDR exists)	Access sysadmin	Unused	Unused
console attach-sdr location 0/RP0 tty-name console1 sdr-name sdr1	Access sysadmin	sdr1	Unused
console attach-sdr location 0/RP0 tty-name console2 sdr-name sdr2	Access sysadmin	sdr1	sdr2
On system reload	Access sysadmin	sdr1	sdr2
console attach-sdr location 0/RP0 tty-name console2 sdr-name sdr3	Access sysadmin	sdr1	sdr3
console attach-sdr location 0/RP0 tty-name console1 sdr-name sdr2	Access sysadmin	sdr2	sdr3
no sdr (all named-sdr removed)	Access sysadmin	Unused	Unused

Command	Console0 Access	Console1 Access to XR VM	Console2 Access to XR VM
console attach-sdr location 0/RP0 tty-name console2 sdr-name sdr1 (assuming only one named-SDR (sdr1) is configured)	Access sysadmin	Unused	sdr1

Multi-SDR Environment

By default, the system boots up in the Single Owner Single Tenant (SOST) mode. This setup is termed as default-SDR. Creating explicit user defined SDRs and assigning inventory to it, causes the system to transit from SOST mode to Single Owner Multiple Tenant (SOMT) mode. This setup is termed as multi-SDR and the user defined SDR is called as named-SDR.

These are the attributes of an multi-SDR setup:

- The system administrator for the admin plane, manages the system inventory and allocates resources to each named-SDR. The RPs and LCs can be allocated or deallocated to a named-SDR without affecting other SDRs in the system. RP resources like CPU and memory, and LC resources are configurable.
- Each named-SDR can be administered, managed, and operated independent of other named-SDRs in the system. Named-SDR is also independent of the admin plane.
- Each named-SDR can be run on a version of Cisco IOS XR software that is independent of the versions running on other named-SDRs and the admin plane.
- Each named-SDR can be upgraded (non-ISSU) independently of other named-SDRs in the system.
- AAA administrator needs to provide permissions to access the admin plane through named-SDR.
- RPs, fabric cards and other system resources are shared across multi-SDRs.
- Line cards cannot be shared among named-SDRs. But, multiple line cards can be allocated to a named-SDR.
- Depending on the named-SDR configuration, each SDR will have its own RP pair. RP Fail Over (RPFO) in a named-SDR, does not impact other SDRs as each SDR has its own RP pair.
- Hard disk is partitioned between the SDRs. Each named-SDR gets ~33GB of the system hard disk.
- XR management traffic in multi-SDR is tagged, which needs to be un-tagged. For more information, see XR Management Traffic in Multi-SDR Environment, on page 114.
- There is no difference between the default-SDR and multi-SDR with respect to the XR features, provisioning of features, or the user interaction.
- Muti-SDR does not support full system upgrade.
- Due to smaller hard disk size, OS images cannot be stored in the hard disk of named-SDR.

- USB is accessible from System Admin VM and XR VM of default-SDR. However, in case of a named-SDR, USB is not accessible from the XR VM and can be accessed only from System Admin VM.
- Secure copy (SCP) operation for below mentioned cases is not supported from Cisco IOS XR Release 6.3.1 onwards:
  - between named-SDR (XR VM) and SysAdmin (System Admin VM)
  - from one named-SDR to another named-SDR

Note

- On-the-fly modification of RP resources (CPU or memory) of a named-SDR is not recommended.
- To convert from SOMT to SOST, the system must be USB booted.

Software Upgrade in Multi-SDR Environment

Each named SDR can be upgraded or downgraded individually. SMUs and packages installed on each SDR is independent of other SDRs. For upgrade/downgraded details, see *System Setup and Software Installation Guide for Cisco NCS 6000 Series Routers*.

Note Full system upgrade is not supported in a multi-SDR setup. The full system upgrade means simultaneous upgrade of the System Admin VM and all named-SDRs. Named-SDRs need to be upgraded individually. Also, only one operation of upgrade, downgrade, SMU install, or SMU deactivation can be performed at a time.

An Orchestrated Calvados Upgrade (OCU) cannot be performed on a Multi-SDR from Release 6.1.x to later releases (Release 6.2.x onwards).

System Admin VM can be upgraded or downgraded independent of named SDRs.

USB Accessibility

USB port is accessible only from System Admin VM. It is not accessible from named-SDR; hence, you cannot install image/SMU/Package from USB in multi-SDR mode.

Note If type 8,9, or 10 is the secret key configured, then before downgrading to 6.6.3 and earlier versions, perform either of the following methods:

- Type a combination of secret type and encrypted key instead of plain text for the password. Example:

  ```
  username root
  group root-lr
  group cisco-support
  secret 10
  $6$Mwaqg/jdBPOn4g/.$PrJP2KjsCbL6bZqmYOej5Ay67S/sSWJNlkiYhCTc/B/35E1kJBqffmBtn.ddQEH0O2CU7V.ZEMmqIg7uE8cfz0
  ```

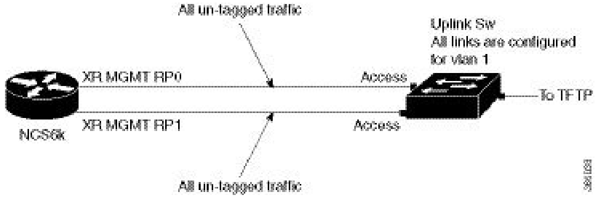
  This is because 6.6.3 and earlier versions do not support type 8,9, or 10 key type.

- Ensure that there are secret type 5 users on the system.

XR Management Traffic in Multi-SDR Environment

Each RP has single physical management port. Hence, management traffic of the named-SDR is tagged with the VLAN-ID that is specified during named-SDR configuration. Tagged management traffic of each named-SDR needs to be segregated by the user using an external switch. As XR VM is unaware of the VLAN tagging or multi-SDR, there is no change in the management port configuration for the XR VM.

Figure 5: Example: XR Management Traffic in Default-SDR Environment



In case of default-SDR, all management traffic is untagged.

Sample Configuration for XR Management Traffic in Multi-SDR Environment

```
sdr INET_RI
pairing-mode  inter-rack
resources card-type RP
  vm-memory 11
  vm-cpu    4
!
location 0/1
!
location 0/2
!
location 1/1
!
location 1/2
!
location 0/RP0
!
location 0/RP1
!
location 1/RP0
!
```

```
location 1/RP1
!
sdr SU_RI
pairing-mode  inter-rack
resources card-type RP
  vm-memory 11
  vm-cpu    4
!
location 0/4
!
location 1/4
!
location 0/5
!
location 1/5
!
location 0/7
!
location 0/RP0
!
location 0/RP1
!
location 1/RP0
!
location 1/RP1
!
sdr VPN_RI
pairing-mode  inter-rack
resources card-type RP
  vm-memory 11
  vm-cpu    4
!
location 0/3
!
location 1/3
!
Location 1/7
!
location 0/6
!
location 1/6
!
location 0/RP0
!
location 0/RP1
!
location 1/RP0
!
location 1/RP1
!
```

CHAPTER 10

Configuring Collapsed Forwarding

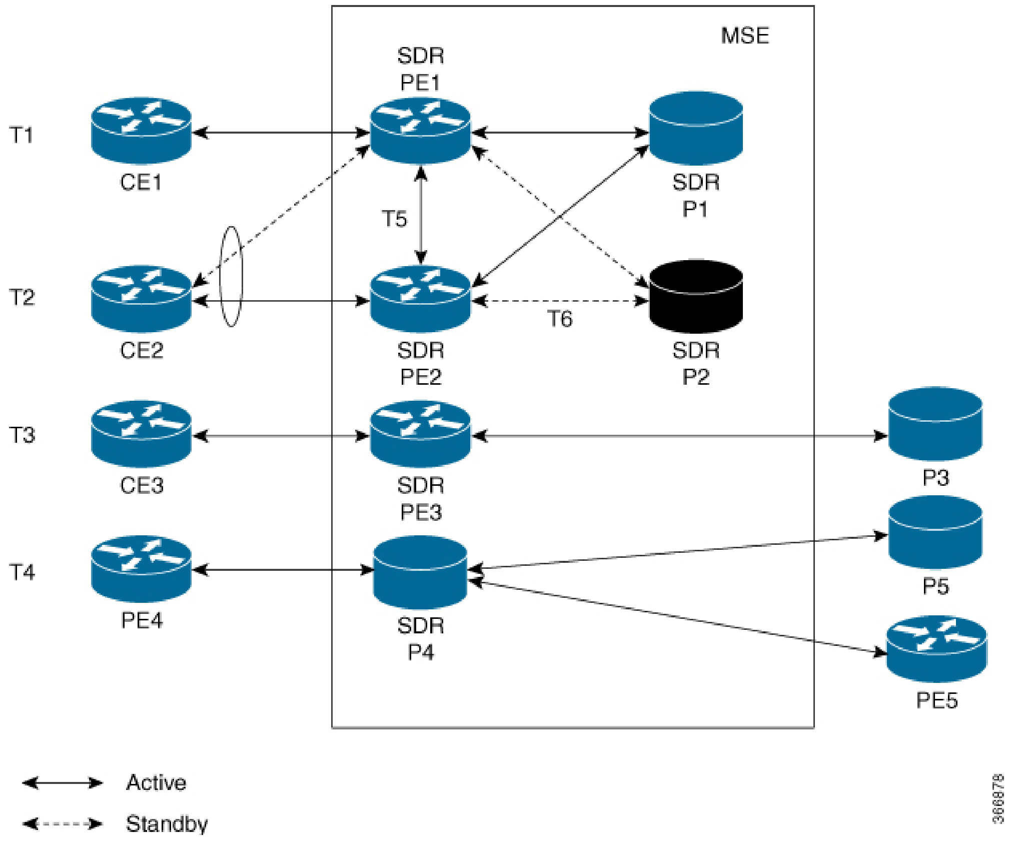
This module contains the following topics:

Overview of Collapsed Forwarding

Cisco multi-switch edge (MSE) solution leverages the capabilities of the Cisco NCS 6000 series router to host multiple logical SDRs which can act as core or edge routers based on how the SDRs are configured.

The following figure specifies all the possible topologies supported by the Cisco MSE solution.

Figure 6: Supported Topologies by Cisco MSE



- Topology T1-In this topology, all the PE SDRs converge to a common core router P1 to achieve statistical multiplexing gain.
- Topology T2- This topology is to support redundancy at the service provider edge (dual homing) or can be also used for load balancing.
- Topology T3 - In this topology, the service edge router converges to the external core router, outside the MSE.
- Topology T4 - In this topology, external edge router converges to the core router inside the MSE.
- Topology T5 - This topology is used for edge router connectivity for terminating the traffic without going to the core internal router.
- Topology T6 - This topology is to support core router redundancy. In case P1 router goes down, P2 as redundant router runs with the same configuration.

For forwarding traffic from one SDR to another SDR, the existing solution was to connect the SDRs using external cables and ports. This approach is not cost effective for the service providers since it reduces the availability of ports for services. To overcome this issue, Cisco MSE solution uses another approach known as collapsed forwarding.

In collapsed forwarding, inter SDR traffic is handled by the internal fabric itself without requiring the external cables. A newly created SDR interface functions as a point-to-point virtual interface, connecting SDR routers in the system. This virtual interface that connects two SDRs to each other is known as cross SDR interconnect (CSI) interface.

Supported Features and Restrictions

This section provides information about the supported features and restrictions for collapsed forwarding configuration.

- The CSI interface is a point-to-point interface that can be configured with IPv4 or IPv6 address.
- Only routing protocols, MPLS, and multicast can be configured over the CSI interface.
- Explicit Binding SID over CSI Interfaces is supported.

  For more information, refer to the "Configure SR-TE Policies" chapter in the *Segment Routing Configuration Guide for Cisco NCS 6000 Series Routers*.
- BGP Egress Peer Engineering (EPE) over CSI interfaces is supported. This support allows a BGP neighbor established over a CSI interface to be allocated a BGP Egress Peer Engineering (EPE) segment.

  For more information, refer to the "Configure Segment Routing for BGP" chapter in the *Segment Routing Configuration Guide for Cisco NCS 6000 Series Routers*.
- IPv4 and IPv6 egress ACLs are not supported on CSI interfaces.
- QoS and NetFlow are not supported over the CSI interface.
- ACL based forwarding, ACL logging, and per interface ACL statistics are not supported on CSI interfaces.
- Sub interfaces of VLANs for CSI interface is not supported.
- For a system, only up to 15 CSI interfaces are supported.

Configuring Collapsed Forwarding

Configuring collapsed forwarding includes the following steps.

1. Create named SDRs using the system administration configuration mode.
2. Configure the CSI interface using the system administration configuration mode.
3. Assign the IP addresses for the CSI interface on the required SDRs from XR configuration mode.
4. Configure the routing protocols between SDR1 and SDR2 over the CSI interface.

Example: Creating Named SDRs

This example shows how to create named SDRs. In this example, two named SDRs, SDR1 and SDR2 are created and RP resources and line cards are allocated to the SDRs.

```
sysadmin-vm:0_RP0(config)# sdr sdr1
sysadmin-vm:0_RP0(config-sdr-sdr1)# resources mgmt_ext_vlan 11
sysadmin-vm:0_RP0(config-sdr-sdr1)# resources card-type RP
sysadmin-vm:0_RP0(config-card-type-RP)# vm-memory 11
sysadmin-vm:0_RP0(config-card-type-RP)# vm-cpu 4
```

```
sysadmin-vm:0_RP0(config-card-type-RP)# location 0/RP0
sysadmin-vm:0_RP0(config-location-0/RP0)# location 0/RP1
sysadmin-vm:0_RP0(config-location-0/RP1)# exit
sysadmin-vm:0_RP0(config-sdr-sdr1)# location 0/0
sysadmin-vm:0_RP0(config-sdr-sdr1)# commit
sysadmin-vm:0_RP0(config)# sdr sdr2
sysadmin-vm:0_RP0(config-sdr-sdr2)# resources mgmt_ext_vlan 12
sysadmin-vm:0_RP0(config-sdr-sdr2)# resources card-type RP
sysadmin-vm:0_RP0(config-card-type-RP)# vm-memory 11
sysadmin-vm:0_RP0(config-card-type-RP)# vm-cpu 4
sysadmin-vm:0_RP0(config-sdr-sdr2)# location 0/RP0
sysadmin-vm:0_RP0(config-location-0/RP0)# location 0/RP1
sysadmin-vm:0_RP0(config-location-0/RP1)# exit
sysadmin-vm:0_RP0(config-sdr-sdr2)# location 0/1
sysadmin-vm:0_RP0(config-sdr-sdr2)# commit
sysadmin-vm:0_RP0# config
sysadmin-vm:0_RP0(config)# console attach-sdr location 0/RP0 tty-name console1 sdr-name
SDR1
sysadmin-vm:0_RP0(config)# console attach-sdr location 0/RP1 tty-name console1 sdr-name
SDR1
sysadmin-vm:0_RP0(config)# console attach-sdr location 0/RP0 tty-name console2 sdr-name
SDR2
sysadmin-vm:0_RP0(config)# console attach-sdr location 0/RP1 tty-name console2 sdr-name
SDR2
sysadmin-vm:0_RP0(config)# commit
```

For detailed information about configuring named SDRs, see Setup Console Access for Named-SDR, on page 109.

Example: Configuring the CSI Interface

This example shows how to configure the CSI interface between two SDRs. When you perform this task, a single point-to-point link is created with one endpoint in SDR1 (called csi1 in SDR1) and the other endpoint in SDR2 (also called csi1 in SDR2). You can configure up to 15 CSI interfaces and the CSI-ID can be a number from 1 to 15.

```
sysadmin-vm:0_RP0(config)# connect sdr sdr1 sdr2 csi-id 1
```

Example: Configuring IP addresses on the CSI Interfaces

Once the CSI interface is created, you need to configure IP addresses for the CSI interface on the inter connected SDRs using the XR configuration mode.

This example shows how to configure IPv4 addresses on the CSI interface on SDR1.

```
RP/0/RP0/CPU1:router(config)# interface csi 1
RP/0/RP0/CPU1:router(config-if)# ipv4 address 1.1.1.1 255.255.255.0
RP/0/RP0/CPU1:router(config-if)# exit
RP/0/RP0/CPU1:router(config)# commit
```

This example shows how to configure IPv4 addresses on the CSI interface on SDR2.

```
RP/0/RP0/CPU2:router(config)# interface csi 1
RP/0/RP0/CPU2:router(config-if)# ipv4 address 1.1.1.2 255.255.255.0
RP/0/RP0/CPU1:router(config-if)# exit
```

Example: Configuring the Routing Protocols

This example shows how to configure a routing protocol over the CSI interface. In this example, IS-IS is used as the routing protocol. You need to configure IS-IS on SDR1 and SDR2.

```
RP/0/RP0/CPU1:router(config)# router isis 1
RP/0/RP0/CPU1:router(config-isis)# is-type level-2-only
RP/0/RP0/CPU1:router(config-isis)# net 49.0001.0001.0001.0001.00
RP/0/RP0/CPU1:router(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU1:router(config-isis-af)# metric-style wide level 1
RP/0/RP0/CPU1:router(config-isis-af)# exit
RP/0/RP0/CPU1:router(config-isis)# interface csi 1
RP/0/RP0/CPU1:router(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU1:router(config-isis-if-af)# exit
RP/0/RP0/CPU1:router(config-isis-if)# exit
```

For more information about configuring the routing protocols including IS-IS on NCS6000, see *Routing Configuration Guide for Cisco NCS 6000 Series Routers*.

Verifying the CSI Interface Configuration

You can verify the CSI interface configuration by using the **ping** command to verify the connectivity to the CSI from the SDR.

This example shows verifying the CSI interface configuration from SDR1 using the **ping** command.

```
RP/0/RP0/CPU1:router# ping 1.1.1.2
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 99/184/431 ms
```

This example shows verifying the CSI interface configuration from SDR2 using the **ping** command.

```
RP/0/RP0/CPU2:router# ping 1.1.1.1
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/125/264 ms
```

COFO Unicast NNH with FRR and BFC

COFO Unicast Neighbor-Next-Hop Forwarding Model

Before Cisco IOS XR Release 6.6.1, in COFO traffic forwarding was based on the next-hop based model, ie, each SDR acts as an independent router interconnected by point-to-point interfaces through the internal fabric. In the next-hop based model each SDR forwards the traffic to all neighboring SDRs before the packet reaches the right destination. As a result, the fabric bandwidth utilization is inefficient.

The issue of bandwidth under utilization is overcome using the NNH (neighbor-next-hop) based forwarding model. NNH is collapsed information of neighbor SDR NNH interface. The collapsed NNH has the platform information which leads the traffic to the correct egress LC or slice of the downstream SDR. Therefore, NNH based model is the most optimal way to forward the intra SDR traffic.

FRR and BFC

From this release onwards, COFO supports Fast Re-route (FRR) and Bundle Fast Convergence (BFC) features.

- FRR—The Fast Re-route feature enables fast traffic recovery upon link or router failure by rerouting the traffic over backup tunnels that bypass failed links or node. These are the FRR types that are supported in this release:

  - IP and LDP FRR—FRR driven by an IGP and LDP protocols.
  - TE FRR—FRR driven by MPLS-TE and RSVP protocols.

- BFC—The Bundle Fast Convergence feature provides the ability to converge bundle members within sub seconds instead of multiple seconds.

Implementation Consideration

These points must be considered before configuring COFO:

- Any VPN disposition traffic from the core to the customer follows the NH based forwarding model with explicit-null enabled.
- Any L2 disposition traffic from the core to the customer follows the NH based forwarding model as L2 label info is not collapsed at the ingress.
- Any L3VPN disposition traffic from the core to the customer follows the NH way of forwarding for connected routes redistributed into BGP.
- The traffic traversing from RSVP tunnels with Ingress SDR as midpoint follows the NH based forwarding model.
- Equal-cost multi-path allows a router to insert more than one path to a destination in the routing table to enable load balancing. While configuring ECMP, the maximum path value should be more than the actual ECMP path of the egress SDR. If not then there will be traffic drops due to mismatch of ECMP path at ingress and egress SDR.
- ISIS incremental SPF configuration is not supported for COFO NNH.
- During a line card OIR (online insertion and removal) and slice shut, the traffic hit is longer even though BFC is enabled.
- Traffic drop is observed when the TE FRR is triggered by the BFD on SU (shared uplink) and the trigger is not notified to a VPN.

Recommendation

Under router ISIS configuration, you should configure advertise link attribute to enable COFO NNH. This is required to enable ISIS to collapse the NNH info.

Configuring LDP FRR for COFO

Configuring collapsed forwarding with tunnel fast re-route includes the following step:

Note IGP and LDP is enabled in the network.

1. Configure a bundle interface under ISIS

Example: Configuring a bundle interface under ISIS

```
configure
router isis 124
 interface Bundle-Ether bundlether1
 address-family ipv4 unicast
 fast-reroute per-prefix
!
```

Verification

Before the LDP or IGP FRR triggers, use the **show cef fast-reroute** command to view the active and standby nodes:

```
router#sh cef fast-reroute

Prefix              Next Hop            Interface
192.168.1.1/32      192.168.10.2          bundlether1
                    192.168.10.8          bundlether2 (!) /*backup node*/
```

The above output shows *bundlether1* as Active node and *bundlether2* as Standy or backup node.

Now verify the FRR after the LDP FRR or IGP FRR trigger.

```
router#sh cef fast-reroute

Prefix              Next Hop            Interface
192.168.1.1/32      192.168.10.2          bundlether1  (!) /*backup node*/
                    192.168.10.8          bundlether2
```

Configuring TE FRR for COFO

Configuring collapsed forwarding with tunnel fast re-route includes the following steps:

Note MPLS TE is enabled in the network.

1. Configure the primary and backup TE tunnel
2. Configure a backup tunnel under the TE configuration

Example: Configuring Primary and Backup MPLS-TE tunel

```
mpls traffic-eng
 interface Bundle-Ether151
  backup-path tunnel-te 2

interface tunnel-te1
 ipv4 unnumbered Loopback0
 autoroute announce
 !
 destination 192.168.1.1
 policy-class 4
 record-route
 path-option 1 dynamic
 path-option 2 explicit name SU-P1
!

interface tunnel-te2
```

```
 ipv4 unnumbered Loopback0
 !
 destination 192.168.1.1
 policy-class 5
 record-route
 path-option 1 explicit name SU-P2-P1
 path-option 2 dynamic
!
```

Verification

Use the **show mpls traffic-eng fast-reroute database** command to verify the FRR trigger:

```
router#show mpls traffic-eng fast-reroute database

LSP midpoint FRR information:
LSP Identifier               Local  Out Intf/        FRR Intf/        Status
                             Label  Label            Label
----------------------------- ------ ---------------- ---------------- -------
192.168.1.1 0 [4]             16006  bundlether1:16011  tt2:Pop         Ready
```

In the above LSP starting at headend 192.168.1.1 traverses through the primary (protected) interface bundleether1. When the bundlether1 goes down, traffic is forwarded via the tunnel-te 2 (backup). This action is a Pop action as this is a next-hop tunnel, i.e, the remote label for the original LSP is popped before being forwarded to P2.

Layer 2 Interface Support on CSI

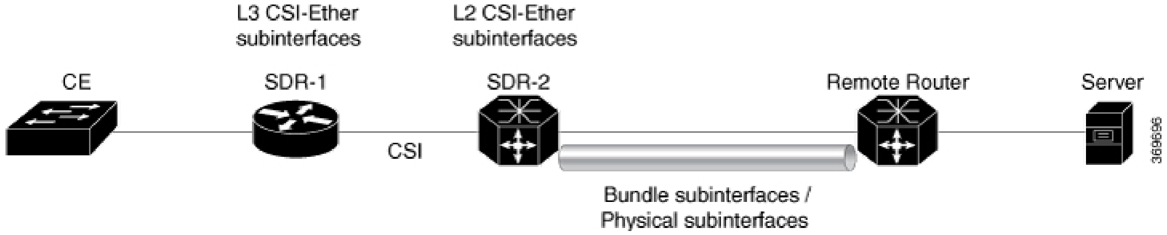
The Layer 2 Interface Support on CSI feature provides a Layer 2 connectivity between the Secure Domain Routers (SDRs). This feature introduces a new virtual interface that is called CSI-Ether interface that enables forwarding of Layer 2 frames between the SDRs.

Some cloud applications in the service provider network, such as the speed test, Internet Protocol Service Level Agreement (IPSLA), and so on, are sensitive to latency and jitter. To avoid extra L3 hops, you can use Layer 2 connectivity between the cloud applications in your network and the customer edge (CE) facing SDRs.

You can interconnect two SDRs through a single point-to-point link using the **connect sdr** command in the sysadmin mode. This creates a CSI-Ether interface along with the CSI interface between these two SDRs. You must use this CSI-Ether interface only as VLAN subinterface. You can configure the dot1q identifier on the CSI-Ether VLAN subinterface. You can create up to 10 dot1q VLAN subinterfaces. The range is from 1 to 10. You must use the same dot1q encapsulation identifier for a given VLAN subinterface on the SDRs and the remote router.

Topology

Figure 7: Layer 2 Interface Support on CSI


L3 CSI-Ether subinterfaces
L2 CSI-Ether subinterfaces
CE
SDR-1
SDR-2
Remote Router
Server
CSI
Bundle subinterfaces / Physical subinterfaces
369596

In this topology, the SDR-2 is a Layer 2 cross-connect and its layer 3 neighborship is between SDR-1 and the remote router. On SDR-2, configure CSI-Ether subinterface as Layer 2 interface towards SDR-1, and configure bundle or physical subinterface as Layer 2 interface towards the remote router. Configure L2 cross-connect between CSI-Ether subinterface and bundle or physical subinterface. On SDR-1, configure CSI-Ether subinterface as Layer 3 interface. Similarly, configure bundle or physical subinterface as Layer 3 interface on the remote router. Configure Layer 3 neighborship between SDR-1 and the remote router using IGP or BGP.

Configure Layer 2 Interface on CSI

Configuration Example

Perform this task to configure this feature.

```
/* Configure SDR-1 */
Router# configure
Router(config)# interface CSI-Ether4.1
Router(config-subif)# ipv4 address 192.0.2.1 255.255.255.0
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# exit
Router(config)# interface CSI-Ether4.2
Router(config-subif)# ipv4 address 198.51.100.1 255.255.255.0
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# commit

/* Configure SDR-2 */
Router# configure
Router(config)# interface CSI-Ether4.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# exit
Router(config)# interface CSI-Ether4.2 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# exit
Router(config)# l2vpn
Router(config-l2vpn)# pw-class mpls
Router(config-l2vpn-pwc)# encapsulation mpls
Router(config-l2vpn-pwc-mpls)# exit
Router(config-l2vpn-pwc)# exit
Router(config-l2vpn)# xconnect group SDR2_SDR1
Router(config-l2vpn-xc)# p2p SDR_SDR
Router(config-l2vpn-xc-p2p)# interface CSI-Ether4.1
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether15221.1
Router(config-l2vpn-xc-p2p)# exit
Router(config-l2vpn-xc)# p2p SDR_SDR1
Router(config-l2vpn-xc-p2p)# interface CSI-Ether4.2
Router(config-l2vpn-xc-p2p)# interface Bundle-Ether15221.2
Router(config-l2vpn-xc-p2p)# commit

/* Configure Remote Router */
Router# configure
Router(config)# interface Bundle-Ether15221
Router(config-if)# mtu 9500
Router(config-if)# exit

Router(config)# interface Bundle-Ether15221.1
Router(config-subif)# ipv4 address 192.0.2.2 255.255.255.0
Router(config-subif)# encapsulation dot1q 1
Router(config)# interface Bundle-Ether15221.2
Router(config-subif)# ipv4 address 198.51.100.2 255.255.255.0
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# exit
```

Running Configuration

This section shows the running configuration of Layer 2 Interface on CSI.

```
/* On SDR-1 */
configure
 interface CSI-Ether4.1
 ipv4 address 192.0.2.1 255.255.255.0
 encapsulation dot1q 1
!
interface CSI-Ether4.2
 ipv4 address 198.51.100.1 255.255.255.0
 encapsulation dot1q 2
!

/* On SDR-2 */
configure
 interface CSI-Ether4.1 l2transport
 encapsulation dot1q 1
!
 interface CSI-Ether4.2 l2transport
 encapsulation dot1q 2
!
l2vpn
 pw-class mpls
  encapsulation mpls
  !
 !
 xconnect group SDR1_SDR2
  p2p SDR_SDR
   interface CSI-Ether4.1
   interface Bundle-Ether15221.1
  !
  p2p SDR_SDR1
   interface CSI-Ether4.2
   interface Bundle-Ether15221.2

/* On Remote Router */
configure
 interface Bundle-Ether15221
  mtu 9500
!

 interface Bundle-Ether15221.1
  ipv4 address 192.0.2.2 255.255.255.0
   encapsulation dot1q 1
!
 interface Bundle-Ether15221.2
  ipv4 address 198.51.100.2 255.255.255.0
   encapsulation dot1q 2
!
!
```

Verification

Verify Layer 2 Interface on CSI configuration.

```
Router:SDR-2# show l2vpn xconnect
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
        SB = Standby, SR = Standby Ready, (PP) = Partially Programmed

XConnect                   Segment 1                      Segment 2
Group        Name      ST   Description             ST      Description             ST
------------------------   ------------------------------   ------------------------------
SDR1_SDR2    SDR_SDR   UP   CE4.1                   UP      BE15221.1               UP
----------------------------------------------------------------------------------------
SDR1_SDR2    SDR_SDR1  UP   CE4.2                   UP      BE15221.2               UP
----------------------------------------------------------------------------------------
```

Related Topics

- Layer 2 Interface Support on CSI , on page 124

Associated Commands

- connect sdr
- show l2vpn xconnect

Configure Layer-2 CSI interface MTU

Configuration Example

Perform this task to configure this feature.

```
Router# configure
Router(config)# interface CSI-Ether1
Router(config-if)# csi-Ether mtu 1500
Router(config-if)# commit
```

You can configure an MTU value for the interface. If sub interfaces are created within the interface, the same MTU is configured on them. For example, MTU of a sub interface CSI-Ether1.1 will be 1500 since it inherits the MTU value of CSI-Ether1.

Running Configuration

```
configure
 interface CSI-Ether1
 csi-Ether mtu 1500
!
```

Verification

```
Router# show running-config interface csi-ether1

interface CSI-Ether1
CSI-Ether mtu 1500
!

Router # show interfaces csi-ether1

CSI-Ether1 is up, line protocol is up
  Interface state transitions: 5
  Hardware is Cross SDR Ethernet, address is 0042.68be.bb00
  Internet address is Unknown
  MTU 1500 bytes, BW 4000000000 Kbit (Max: 4000000000 Kbit)
.
```

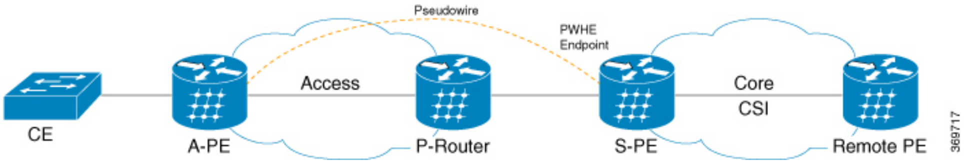
Layer 2 Interface Support on Pseudowire Headend

The Layer 2 Interface Support on Pseudowire Headend feature provides a virtual Layer 2 interface on a pseudowire (PW) for a service provider edge (PE) router. This feature allows termination of access pseudowires (PW) into an L2 domain. This feature allows you to send the L2 traffic over PWs from the access side to the core side. You can provision Lawful Intercept (LI) on a per PWHE interface basis, on a service provider edge (PE) router along with the regular Layer 2 services. This feature reduces the capital expenditure in access and aggregation network. This feature helps the customer network to distribute and scale the customer facing Layer 2 UNI interface set.

Restrictions

- The generic interface list supports only main interfaces, and not subinterfaces.
- Subinterfaces support only bridged interworking (VC type 5) mode.
- You can have a maximum of eight members in the generic interface list.
- There must be a reachability between a given pair of Access Provider Edge (A-PE) and Service Provider Edge (S-PE) apart from PWHE Interface.
- You can configure only one PWHE attachment circuit (AC) on a bridge. However, all 2K bridges can have one PWHE AC.

Figure 8: Layer 2 Support on Pseudowire Headend



Consider a topology where you enable PWHE on S-PE. Create an L2VPN Xconnect from A-PE (interface that connects to CE) to the PWHE interface that is created on S-PE. Configure EVPN or VPLS on S-PE to reach the remote PE. When traffic from CE reaches the A-PE, the A-PE forwards the frames to the PWHE interface on the S-PE. S-PE sends the traffic to the remote PE based on the VLAN configuration.

Configure Layer 2 Interface on Pseudowire Headend

Configuration Example

This section describes how you can configure Layer 2 Interface on Pseudowire Headend on S-PE, A-PE, and remote router.

S-PE Configuration

```
/* S-PE Configuration */

/* Configure generic interface list for PWHE interface and attach the generic interface
list with a PWHE interface. */
Router# configure
```

```
Router(config)# generic-interface-list gil1
Router(config-gen-if-list)# interface Bundle-Ether200
Router(config-gen-if-list)# exit

Router(config)# interface PW-Ether1
Router(config-if)# attach generic-interface-list gil1

/* Configure Layer 2 transport and PW class for PWHE interface */

Router# configure
Router(config)# interface PW-Ether1.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# mtu 1514
Router(config-subif)# service-policy input ingress-parent
Router(config-subif)# exit
Router(config)# l2vpn
Router(config-l2vpn)#  pw-class pwe
Router(config-l2vpn-pwc)# encapsulation mpls
Router(config-l2vpn-pwc-mpls)# control-word
Router(config-l2vpn-pwc-mpls)# transport-mode ethernet

/* Configure cross-connect for PWHE interface */

Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xcpw1
Router(config-l2vpn-xc)# p2p 1
Router(config-l2vpn-xc-p2p)# interface PW-Ether1
Router(config-l2vpn-xc-p2p)# neighbor ipv4 192.0.2.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# pw-class pwe

/* Configure the bridge domain */

Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface PW-Ether1.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# vfi vf
Router(config-l2vpn-bg-bd-vfi)# neighbor 198.51.100.1 pw-id 1
```

A-PE Configuration

```
/* A-PE Configuration */

/* Configure PWHE  Ethernet interface */

Router# configure
Router(config)# interface HundredGigE0/3/0/2.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# mtu 1514

/* Configure cross-connect for PWHE interface */

Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group xcpw1
Router(config-l2vpn-xc)# p2p 1
Router(config-l2vpn-xc-p2p)# interface HundredGigE0/3/0/2.1
Router(config-l2vpn-xc-p2p)# neighbor ipv4 203.0.113.1 pw-id 1
Router(config-l2vpn-xc-p2p-pw)# pw-class pwe
```

Remote Router Configuration

```
/* Configure PWHE  Ethernet interface */
Router# configure
Router(config)# interface HundredGigE0/5/0/1.1 l2transport
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# mtu 1514

/* Configure the bridge domain */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd1
Router(config-l2vpn-bg-bd)# interface HundredGigE0/5/0/1.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# vfi vf
Router(config-l2vpn-bg-bd-vfi)# neighbor 203.0.113.1 pw-id 1
```

Running Configuration

This section shows Layer 2 Interface on Pseudowire Headend running configuration.

```
/* On S-PE */
configure
 generic-interface-list gil1
  interface Bundle-Ether200
!
 interface PW-Ether1
  attach generic-interface-list gil1
!
configure
 interface PW-Ether1.1 l2transport
  encapsulation dot1q 1
  mtu 1514
  service-policy input ingress-parent

l2vpn
 pw-class pwe
  encapsulation mpls
   control-word
   transport-mode ethernet
!
!
l2vpn
 xconnect group xcpw1
  p2p 1
   interface PW-Ether1
   neighbor ipv4 203.0.113.1 pw-id 1 pw-id 1
   pw-class pwe
!
l2vpn
 bridge group bg1
  bridge-domain bd1
   interface interface PW-Ether1.1
!
   vfi vf
    neighbor 198.51.100.1 pw-id 1
```

```
/* On A-PE */

configure
```

```
 interface HundredGigE0/3/0/2.1 l2transport
  encapsulation dot1q 1
  mtu 1514
!

l2vpn
 xconnect group xcpw1
  p2p 1
   interface HundredGigE0/3/0/2.1
    neighbor ipv4 3.3.3.3 pw-id 1
    pw-class pwe
  !
!

l2vpn
 xconnect group APE2-PE1-PORT
  p2p APE2-PE1-5001
   interface TenGigE0/0/0/0
   neighbor ipv4 100.1.1.1 pw-id 5001
    pw-class APE2-PE1-PORT
   !
!
```

```
/* On Remote Router */

interface HundredGigE0/5/0/1.1 l2transport
 encapsulation dot1q 1
 mtu 1514
!

l2vpn
 bridge group bg1
  bridge-domain bd1
   interface HundredGigE0/5/0/1.1
!
   vfi vf
    neighbor 203.0.113.1 pw-id 1
```

Verification

The show outputs given in the following section display the details of the configuration of PW Ethernet interface and cross-connect, and the status of their configuration on S-PE and A-PE.

```
/* S-PE Configuration */

Router-S-PE# show l2vpn xconnect summary

Mon Jul 15 07:25:34.504 UTC
Number of groups: 4000
Number of xconnects: 4000
Up: 4000 Down: 0 Unresolved: 0 Partially-programmed: 0
AC-PW: 4000 AC-AC: 0 PW-PW: 0 Monitor-Session-PW: 0
Number of Admin Down segments: 0
Number of MP2MP xconnects: 0
Up 0 Down 0
Advertised: 0 Non-Advertised: 0
Number of CE Connections: 0
Advertised: 0 Non-Advertised: 0
Backup PW:
Configured : 0
UP : 0
```

```
Down : 0
Admin Down : 0
Unresolved : 0
Standby : 0
Standby Ready: 0
Backup Interface:
Configured : 0
UP : 0
Down : 0
Admin Down : 0
Unresolved : 0
Standby : 0


Router-S-PE# show l2vpn bridge-domain summary

Mon Jul 15 07:26:27.388 UTC
Number of groups: 1, VLAN switches: 0
Number of bridge-domains: 2001, Up: 2001, Shutdown: 0, Partially-
programmed: 0
Default: 2001, pbb-edge: 0, pbb-core: 0
Number of ACs: 2001 Up: 2001, Down: 0, Partially-programmed: 0
Number of PWs: 2001 Up: 2001, Down: 0, Standby: 0, Partially-programmed: 0
Number of P2MP PWs: 0, Up: 0, Down: 0, other-state: 0
Number of VNIs: 0, Up: 0, Down: 0, Unresolved: 0
```

```
/* A-PE configuration details */

Router-A-PE#show l2vpn xconnect summary

Mon Jul 15 07:23:54.838 UTC
Number of groups: 4001
Number of xconnects: 4001
Up: 4000 Down: 0 Unresolved: 0 Partially-programmed: 0
AC-PW: 4001 AC-AC: 0 PW-PW: 0 Monitor-Session-PW: 0
Number of Admin Down segments: 0
Number of MP2MP xconnects: 0
Up 0 Down 0
Advertised: 0 Non-Advertised: 0
Number of CE Connections: 0
Advertised: 0 Non-Advertised: 0
Backup PW:
Configured : 0
UP : 0
Down : 0
Admin Down : 0
Unresolved : 0
Standby : 0
Standby Ready: 0
Backup Interface:
Configured : 0
UP : 0
Down : 0
Admin Down : 0
Unresolved : 0
Standby : 0
```

Related Topics

- Layer 2 Interface Support on Pseudowire Headend, on page 128

Associated Commands

- show l2vpn xconnect
- show l2vpn bridge-domain

Layer 2 Support for LI and QoS on Pseudowire Headend Interface

The Layer 2 Support for LI and QoS on Pseudowire Headend Interface feature allows you to enable Lawful Intercept (LI) and Quality of Service (QoS) on PWHE L2 subinterface.

Lawful Intercept

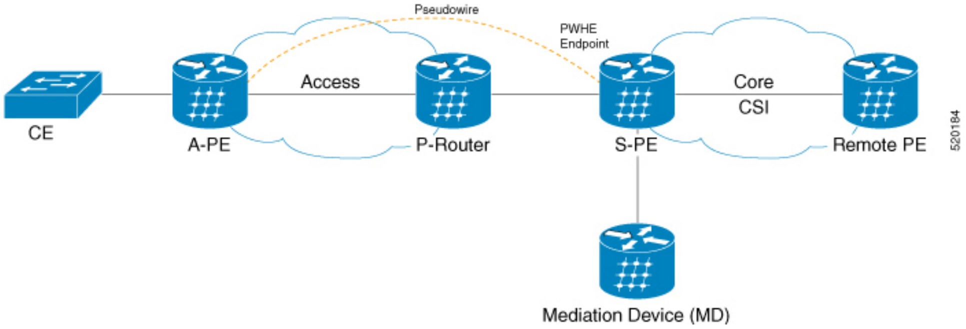
The Lawful Intercept (LI) feature supports service providers in meeting the requirements of law enforcement agencies to provide the ability to intercept Voice-over-Internet protocol (VoIP) or data traffic going through the edge routers. This feature allows you to replicate and forward intercepted packets to the mediation device (MD).

LI is the process by which law enforcement agencies conduct electronic surveillance of circuit and packet-mode communications, authorized by judicial or administrative order. Service providers worldwide are legally required to assist law enforcement agencies in conducting electronic surveillance in both circuit-switched and packet-mode networks.

Only authorized service provider personnel are permitted to process and configure lawfully authorized intercept orders. Network administrators and technicians are prohibited from obtaining knowledge of lawfully authorized intercept orders, or intercepts in progress. Error messages or program messages for intercepts installed in the router are not displayed on the console.

Consider a topology where you enable LI on S-PE. Connect S-PE to the mediation device (MD). Intercept the incoming and outgoing traffic on S-PE and forward it to the MD.

Figure 9: Lawful Intercept



Configure Lawful Interface on PWHE Interface

Perform the following tasks to configure Lawful Interface (LI) on PWHE interface:

- Configure SNMP server
- Configure MD
- Create a Tap

- Enable a Tap
- Disable a Tap
- Destroy a Tap
- Destroy MD

Configuration Example

Perform the following task to configure SNMP server on S-PE. The SNMP server configuration allows the MD to intercept VoIP or data sessions.

```
/* S-PE Configuration */

Router# configure
Router(config)# snmp-server engineID local 80:00:00:09:03:00:00:11:92:02:9D:06
Router(config)# snmp-server community public RW SDROwner
Router(config)# snmp-server user CharlieDChief  li-group v3 auth md5 clear lab priv des56
clear lab SDROwner
Router(config)# snmp-server view li-view ciscoTap2MIB included
Router(config)# snmp-server view li-view ciscoIpTapMIB included
Router(config)# snmp-server view li-view ciscoUserConnectionTapMIB included
Router(config)# snmp-server view li-view snmp included
Router(config)# snmp-server view li-view ifMIB included
Router(config)# snmp-server view li-view system included
Router(config)# snmp-server view li-view 1.3.6.1.2.1 included
Router(config)# snmp-server group li-group v3 priv read li-view write li-view notify li-view


/* MD Configuration */
The following configuration must be sent from any linux server or router from where management
 IP address is reachable.

./setany -v3 4.16.7.25 CharlieDChief \
 cTap2MediationDestAddressType.1 ipv4 \
 cTap2MediationDestAddress.1 "47 47 47 02" \
 cTap2MediationDestPort.1 50000 \
 cTap2MediationSrcInterface.1 00 \
 cTap2MediationTransport.1 udp \
 cTap2MediationTimeout.1 "07 E3 09 1B 10 00 00 00 2B 05 0E" \
 cTap2MediationNotificationEnable.1 true \
 cTap2MediationStatus.1 createAndGo

/* Tap Creation */
./setany -v3 4.16.7.25 CharlieDChief citapStreamInterface.1.1 4076 citapStreamStatus.1.1
createAndGo

4076 -> Index of the PWHE sub interface where you apply taps.
4.16.7.25 -> Management IP address of the router.

/* Enable a Tap */
./setany -v3 4.16.7.25 CharlieDChief cTap2StreamType.1.1 ip cTap2StreamInterceptEnable.1.1
 true cTap2StreamStatus.1.1 createAndGo

/* Disable a Tap */
 ./setany -v3 4.16.7.25 CharlieDChief cTap2StreamStatus.1.1 destroy

/* Destroy a Tap */
./setany -v3 4.16.7.25 CharlieDChief citapStreamStatus.1.1 destroy
```

```
/* Destroy MD */
./setany -v3 4.16.7.25 CharlieDChief cTap2MediationStatus.1 destroy
```

Running Configuration

This section shows the Lawful Interface (LI) running configuration.

```
/* S-PE Configuration*/
snmp-server engineID local 80:00:00:09:03:00:00:11:92:02:9D:06
    snmp-server community public RW SDROwner
    snmp-server user CharlieDChief  li-group v3 auth md5 clear lab priv des56 clear lab
SDROwner
    snmp-server view li-view ciscoTap2MIB included
    snmp-server view li-view ciscoIpTapMIB included
    snmp-server view li-view ciscoUserConnectionTapMIB included
    snmp-server view li-view snmp included
    snmp-server view li-view ifMIB included
    snmp-server view li-view system included
    snmp-server view li-view 1.3.6.1.2.1 included
    snmp-server group li-group v3 priv read li-view write li-view notify li-view

/* MD Configuration */
./setany -v3 4.16.7.25 CharlieDChief \
 cTap2MediationDestAddressType.1 ipv4 \
 cTap2MediationDestAddress.1 "47 47 47 02" \
 cTap2MediationDestPort.1 50000 \
 cTap2MediationSrcInterface.1 00 \
 cTap2MediationTransport.1 udp \
 cTap2MediationTimeout.1 "07 E3 09 1B 10 00 00 00 2B 05 0E" \
 cTap2MediationNotificationEnable.1 true \
 cTap2MediationStatus.1 createAndGo
```

Related Topics

- Layer 2 Interface Support on CSI , on page 124
- Lawful Intercept, on page 133

Quality of Service

Quality of Service (QoS) is the technique of prioritizing traffic flows and providing preferential forwarding for higher-priority packets. You can apply only two-level hierarchical policy over the PWHE interface. The top-level parent policy with a default class is configured with shape or police rate in absolute bandwidth. Use this absolute rate as reference for the child policy where you specify actual match or action.

- On Ingress PWHE subinterface, the classification is performed using L2 fields.
- The QoS feature supports only policing and remarking on PWHE L2 subinterfaces.
- This feature does not support queuing related features such as shape, queue limit.
- The QoS feature does not support egress QoS.

Configure QoS on PWHE Interface

Perform the following task to configure QoS on PWHE interface.

Configuration Example

```
Router# configure
Router(config)# class-map match-any vpl_known
Router(config-cmap)# match vpls known
Router(config-cmap)# end-class-map
Router(config-cmap)# exit

Router(config)# class-map match-any vpl_unknown
Router(config-cmap)# match vpls unknown
Router(config-cmap)# end-class-map
Router(config-cmap)# exit

Router(config)# class-map match-all vpl_broadcast
Router(config-cmap)# match vpls broadcast
Router(config-cmap)# match cos 7
Router(config-cmap)# end-class-map
Router(config-cmap)# exit

Router(config)# class-map match-any vpl_multicast
Router(config-cmap)# match vpls multicast
Router(config-cmap)# end-class-map
Router(config-cmap)# exit

Router(config)# class-map match-all L2_Para
Router(config-cmap)# match source-address mac 0000.0100.0001
Router(config-cmap)# match not dei 1
Router(config-cmap)# match not cos 7
Router(config-cmap)# end-class-map
Router(config-cmap)# exit

Router(config)# policy-map pw-l2-par-ingress
Router(config-pmap)# class class-default
Router(config-pmap-c)# service-policy pw-l2-child-ingress-1
Router(config-pmap-c)# police rate 100 mbps
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# end-policy-map

Router(config)# policy-map pw-l2-child-ingress-1
Router(config-pmap)#  class L2_Para
Router(config-pmap-c)# police rate percent 70 peak-rate percent 80
Router(config-pmap-c-police)# conform-action set mpls experimental imposition 6
Router(config-pmap-c-police)# exceed-action set mpls experimental imposition 4
Router(config-pmap-c-police)# violate-action set mpls experimental imposition 5
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# class vpl_multicast
Router(config-pmap-c)# set qos-group 20
Router(config-pmap-c)# police rate percent 10
Router(config-pmap-c-police)# exit

Router(config-pmap)# class vpl_unknown
Router(config-pmap-c)# set qos-group 200
Router(config-pmap-c)# police rate percent 5
Router(config-pmap-c-police)# exit

Router(config-pmap)# class vpl_broadcast
Router(config-pmap-c)# set qos-group 3
Router(config-pmap-c)# police rate percent 5
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# priority level 1
Router(config-pmap-c)# exit

Router(config-pmap)# class vpl_known
Router(config-pmap-c)# set qos-group 101
Router(config-pmap-c)# police rate percent 10
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# exit
Router(config-pmap-c)# end-policy-map

/* Attach policy to PWHE L2 subinterface */
Router(config)# interface PW-Ether2.1 l2transport
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# mtu 1514
Router(config-subif)# service-policy input pw-l2-par-ingress
Router(config-subif)# commit
```

Running Configuration

This section shows the QoS running configuration.

```
class-map match-any vpl_known
 match vpls known
 end-class-map
!
class-map match-any vpl_unknown
 match vpls unknown
 end-class-map
!
class-map match-all vpl_broadcast
 match vpls broadcast
 match cos 7
 end-class-map
!
class-map match-any vpl_multicast
 match vpls multicast
 end-class-map
!
class-map match-all L2_Para
 match source-address mac 0000.0100.0001
 match not dei 1
 match not cos 7
 end-class-map
!
policy-map pw-l2-par-ingress
 class class-default
  service-policy pw-l2-child-ingress-1
  police rate 100 mbps
  !
 !
 end-policy-map
!
policy-map pw-l2-child-ingress-1
 class L2_Para
  police rate percent 70 peak-rate percent 80
   conform-action set mpls experimental imposition 6
   exceed-action set mpls experimental imposition 4
   violate-action set mpls experimental imposition 5
  !
 !
 class vpl_multicast
  set qos-group 20
  police rate percent 10
  !
 !
interface PW-Ether2.1 l2transport
 encapsulation dot1q 2
 mtu 1514
 service-policy input pw-l2-par-ingress
!
```

Verification

Verify that you have successfully configured QoS on PWHE interface.

```
Router#show policy-map interface pw-ether 2000.1
PW-Ether2000.1 input: pw-l2-par-ingress

Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :            32916025/17313829150          0
    Transmitted         :            32916025/17313829150          0
    Total Dropped       :                   0/0                    0
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :            32916025/17313829150          0
    Policed(exceed)     :                   0/0                    0
    Policed(violate)    :                   0/0                    0
    Policed and dropped :                   0/0

  Policy pw-l2-child-ingress-1 Class L2_Para
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :            32916025/17313829150          0
      Transmitted         :            32916025/17313829150          0
      Total Dropped       :                   0/0                    0
    Policing statistics                (packets/bytes)     (rate - kbps)
      Policed(conform)    :            32916025/17313829150          0
      Policed(exceed)     :                   0/0                    0
      Policed(violate)    :                   0/0                    0
      Policed and dropped :                   0/0
      Policed and dropped(parent policer)  : 0/0

  Policy pw-l2-child-ingress-1 Class vpls_multicast
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :                   0/0                    0
      Transmitted         :                   0/0                    0
      Total Dropped       :                   0/0                    0
    Policing statistics                (packets/bytes)     (rate - kbps)
      Policed(conform)    :                   0/0                    0
      Policed(exceed)     :                   0/0                    0
      Policed(violate)    :                   0/0                    0
      Policed and dropped :                   0/0
      Policed and dropped(parent policer)  : 0/0

  Policy pw-l2-child-ingress-1 Class vpls_unknown
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :                   0/0                    0
      Transmitted         :                   0/0                    0
      Total Dropped       :                   0/0                    0
    Policing statistics                (packets/bytes)     (rate - kbps)
      Policed(conform)    :                   0/0                    0
      Policed(exceed)     :                   0/0                    0
      Policed(violate)    :                   0/0                    0
      Policed and dropped :                   0/0
      Policed and dropped(parent policer)  : 0/0
  Policy pw-l2-child-ingress-1 Class vpls_broadcast
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :                   0/0                    0
      Transmitted         :                   0/0                    0
      Total Dropped       :                   0/0                    0
    Policing statistics                (packets/bytes)     (rate - kbps)
      Policed(conform)    :                   0/0                    0
      Policed(exceed)     :                   0/0                    0
      Policed(violate)    :                   0/0                    0
      Policed and dropped :                   0/0
      Policed and dropped(parent policer)  : 0/0

  Policy pw-l2-child-ingress-1 Class vpls_known
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :                   0/0                    0
      Transmitted         :                   0/0                    0
      Total Dropped       :                   0/0                    0
    Policing statistics                (packets/bytes)     (rate - kbps)
      Policed(conform)    :                   0/0                    0
      Policed(exceed)     :                   0/0                    0
      Policed(violate)    :                   0/0                    0
      Policed and dropped :                   0/0
      Policed and dropped(parent policer)  : 0/0

  Policy pw-l2-child-ingress-1 Class class-default
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :                   0/0                    0
      Transmitted         :                   0/0                    0
      Total Dropped       :                   0/0                    0
PW-Ether2000.1 direction output: Service Policy not installed
RP/B0/CB0/CPU5:CVT-MC-VPN1#sh qos
qos  qos-lib  qos-ma
RP/B0/CB0/CPU5:CVT-MC-VPN1#sh qos ?
  aggregate-bundle-mode  aggreate bundle mode
  ea                     QoS EA show commands(cisco-support)
  inconsistency          QoS inconsistency information
  interface              For interface related QoS information
  rm                     PSE QoS Resource Manager information
  status                 Display status of the service-policy applied on interface (nv
submode)
RP/B0/CB0/CPU5:CVT-MC-VPN1#sh qos interface pw-ether 2000.1
% Incomplete command.
RP/B0/CB0/CPU5:CVT-MC-VPN1#sh qos interface pw-ether 2000.1 input
NOTE:- Configured values are displayed within parentheses
Node 0/1/CPU5, Interface PW-Ether2000.1 Ifh 0x88151456 (PWHE Main) -- input policy
NPU Id:       1
Total number of classes:       7
Interface Bandwidth:           100000000 kbps
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------
Level1 Class                             =   class-default

Policer Bucket Id                        =   0x9000117ea5012
Policer committed rate                   =   100032 kbps (100 mbits/sec)
Policer conform burst                    =   1245184 bytes (default)
Policer conform action                   =   Just TX
Policer exceed action                    =   DROP PKT

   Level2 Class                             =   L2_Para

   Policer Bucket Id                        =   0x9000117eb5012
   Policer committed rate                   =   70016 kbps (70 %)
   Policer peak rate                        =   80064 kbps (80 %)
   Policer conform burst                    =   868352 bytes (default)
   Policer exceed burst                     =   993280 bytes (default)
   Policer conform action                   =   SET IMPOSITION EXP AND TX
   Policer conform action value             =   6
   Policer exceed action                    =   SET IMPOSITION EXP AND TX
   Policer exceed action value              =   4
   Policer violate action                   =   SET IMPOSITION EXP AND TX
   Policer violate action value             =   5

   Level2 Class                             =   vpls_multicast
   New qos group                            =   20

   Policer Bucket Id                        =   0x9000117ec5012
   Policer committed rate                   =   10016 kbps (10 %)
   Policer conform burst                    =   124928 bytes (default)
   Policer conform action                   =   Just TX
   Policer exceed action                    =   DROP PKT

   Level2 Class                             =   vpls_unknown
   New qos group                            =   200

   Policer Bucket Id                        =   0x9000117ed5012
   Policer committed rate                   =   4992 kbps (5 %)
   Policer conform burst                    =   62464 bytes (default)
   Policer conform action                   =   Just TX
   Policer exceed action                    =   DROP PKT

   Level2 Class                             =   vpls_broadcast
   New qos group                            =   3

   Policer Bucket Id                        =   0x9000117ee5012
   Policer committed rate                   =   4992 kbps (5 %)
   Policer conform burst                    =   9216 bytes (default)
   Policer conform action                   =   Just TX
   Policer exceed action                    =   DROP PKT

   Level2 Class                             =   vpls_known
   New qos group                            =   101

   Policer Bucket Id                        =   0x9000117ef5012
   Policer committed rate                   =   10016 kbps (10 %)
   Policer conform burst                    =   124928 bytes (default)
   Policer conform action                   =   Just TX
   Policer exceed action                    =   DROP PKT

   Level2 Class                             =   class-default
   Policer not configured for this class

Node 0/7/CPU5, Interface PW-Ether2000.1 Ifh 0x88151456 (PWHE Main) -- input policy
NPU Id:       2
Total number of classes:       7
Interface Bandwidth:           100000000 kbps
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------
Level1 Class                             =   class-default

Policer Bucket Id                        =   0x9000117ea5022
Policer committed rate                   =   100032 kbps (100 mbits/sec)
Policer conform burst                    =   1245184 bytes (default)
Policer conform action                   =   Just TX
Policer exceed action                    =   DROP PKT

   Level2 Class                             =   L2_Para

   Policer Bucket Id                        =   0x9000117eb5022
   Policer committed rate                  =   70016 kbps (70 %)
   Policer peak rate                       =   80064 kbps (80 %)
   Policer conform burst                   =   868352 bytes (default)
   Policer exceed burst                    =   993280 bytes (default)
   Policer conform action                  =   SET IMPOSITION EXP AND TX
   Policer conform action value            =   6
   Policer exceed action                   =   SET IMPOSITION EXP AND TX
   Policer exceed action value             =   4
   Policer violate action                  =   SET IMPOSITION EXP AND TX
   Policer violate action value            =   5

   Level2 Class                            =   vpls_multicast
   New qos group                           =   20

   Policer Bucket Id                       =   0x9000117ec5022
   Policer committed rate                  =   10016 kbps (10 %)
   Policer conform burst                   =   124928 bytes (default)
   Policer conform action                  =   Just TX
   Policer exceed action                   =   DROP PKT

   Level2 Class                            =   vpls_unknown
   New qos group                           =   200

   Policer Bucket Id                       =   0x9000117ed5022
   Policer committed rate                  =   4992 kbps (5 %)
   Policer conform burst                   =   62464 bytes (default)
   Policer conform action                  =   Just TX
   Policer exceed action                   =   DROP PKT

   Level2 Class                            =   vpls_broadcast
   New qos group                           =   3

   Policer Bucket Id                       =   0x9000117ee5022
   Policer committed rate                  =   4992 kbps (5 %)
   Policer conform burst                   =   9216 bytes (default)
   Policer conform action                  =   Just TX
   Policer exceed action                   =   DROP PKT

   Level2 Class                            =   vpls_known
   New qos group                           =   101

   Policer Bucket Id                       =   0x9000117ef5022
   Policer committed rate                  =   10016 kbps (10 %)
   Policer conform burst                   =   124928 bytes (default)
   Policer conform action                  =   Just TX
   Policer exceed action                   =   DROP PKT

   Level2 Class                            =   class-default
   Policer not configured for this class
```

Related Topics

- Layer 2 Interface Support on CSI , on page 124
- Layer 2 Support for LI and QoS on Pseudowire Headend Interface, on page 133

Associated Commands

- show policy-map

Segment Routing Support on CSI

Table 15: Feature History Table

Feature Name	Release Information	Feature Description
Explicit Binding SID over CSI Interfaces	Release 7.4.1	This feature allows you to configure an explicit binding SID (BSID) over a CSI interface.
BGP Egress Peer Engineering (EPE) over CSI interfaces	Release 7.4.1	This feature allows a BGP neighbor established over a CSI interface to be allocated a BGP Egress Peer Engineering (EPE) segment.

Explicit Binding SID and BGP Egress Peer Engineering (EPE) over CSI interfaces are supported.

BGP Egress Peer engineering (EPE)

Example:

```
Router-SU12# show bgp egress-engineering peers

Egress Engineering Object: 101.18.16.2/32 (0x7fb070ba1e80)
       EPE Type: Peer
        Nexthop: 101.18.16.2
        Version: 521, rn_version: 521
          Flags: 0x00000026
      Local ASN: 100
     Remote ASN: 65001
      Local RID: 100.18.1.1
     Remote RID: 100.16.1.1
  Local Address: 101.18.16.1
      First Hop: 101.18.16.2
           NHID: 0
            IFH: 0x880c0024
          Label: 278193, Refcount: 4
        rpc_set: 0x7fb034099928, ID: 1


Router-SU12# show ipv4 interface brief | i 101.18.16
Thu Jul  1 03:58:39.953 UTC
CSI8                           101.18.16.1     Up              Up       default

Router-SU12# show bgp sessions | i 101.18.16.2
Thu Jul  1 03:59:18.661 UTC
101.18.16.2     default                 0 65001     0     0  Established  NSR Ready
```

CHAPTER 11

Configure EVPN on Collapsed Forwarding

This module contains the following topics:

EVPN on Collapsed Forwarding Overview

Ethernet VPN (EVPN) is a next generation solution that provide Ethernet multipoint services over MPLS networks. EVPN operates in contrast to the existing Virtual Private LAN Service (VPLS) by enabling control-plane based MAC learning in the core. In EVPN, PE's participating in the EVPN instances learn customer MAC routes in Control-Plane using MP-BGP protocol. Control-plane MAC learning brings a number of benefits that allow EVPN to address the VPLS shortcomings, including support for multi-homing with per-flow load balancing. Dual-homing mode in EVPN Multihoming is not supported.

EVPN supports collapsed forwarding. EVPN on non-collapse forwarding mode is not supported. For more information about collapsed forwarding, see *Configuring Collapsed Forwarding* chapter.

EVPN Operation

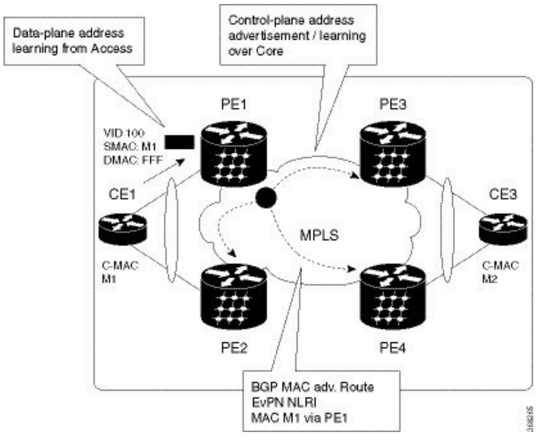
At startup, PEs exchange EVPN routes in order to advertise the following:

- **VPN membership**: The PE discovers all remote PE members of a given EVI. In the case of a multicast ingress replication model, this information is used to build the PE's flood list associated with an EVI.
- **Ethernet segment reachability**: In multi-home scenarios, the PE auto-discovers remote PE and their corresponding redundancy mode (all-active or single-active). In case of segment failures, PEs withdraw

the routes used at this stage in order to trigger fast convergence by signaling a MAC mass withdrawal on remote PEs.

- **Redundancy Group membership**: PEs connected to the same Ethernet segment (multi-homing) automatically discover each other and elect a Designated Forwarder (DF) that is responsible for forwarding Broadcast, Unknown unicast and Multicast (BUM) traffic for a given EVI.

Figure 10: EVPN Operation



EVPN can operate in single homing mode. When EVPN is enabled on PE, routes are advertised where each PE discovers all other member PEs for a given EVPN instance. When an unknown unicast (or BUM) MAC is received on the PE, it is advertised as EVPN type-2 routes to other PEs. MAC routes are advertised to the other PEs using EVPN type-2 routes. In multi-homing scenarios Type 1, 3 and 4 are advertised to discover other PEs and their redundancy modes (single active or active-active). Use of Type-1 route is to auto-discover other PE which hosts the same CE. The other use of this route type is to fast route unicast traffic away from a broken link between CE and PE. Type-4 route is used for electing designated forwarder. For instance, consider the topology when customer traffic arrives at the PE, EVPN MAC advertisement routes distribute reachability information over the core for each customer MAC address learned on local Ethernet segments. Each EVPN MAC route announces the customer MAC address and the Ethernet segment associated with the port where the MAC was learned from and is associated MPLS label. This EVPN MPLS label is used later by remote PEs when sending traffic destined to the advertised MAC address.

EVPN Route Types

The EVPN network layer reachability information (NLRI) provides different route types.

Table 16: EVPN Route Types

Route Type	Name	Usage
1	Ethernet Auto-Discovery (AD) Route	Few routes sent per ES, carry the list of EVIs that belong to ES
2	MAC/IP Advertisement Route	Advertise MAC, address reachability, advertise IP/MAC binding

Route Type	Name	Usage
3	Inclusive Multicast Ethernet Tag Route	Multicast Tunnel End point discovery
4	Ethernet Segment Route	Redundancy group discovery, DF election

Route Type 1: Ethernet Auto-Discovery (AD) Route

The Ethernet (AD) routes are advertised on per EVI and per ESI basis. These routes are sent per ES. They carry the list of EVIs that belong to the ES. The ESI field is set to zero when a CE is single-homed.

Route Type 2: MAC/IP Advertisement Route

The host's IP and MAC addresses are advertised to the peers within NRLI. The control plane learning of MAC addresses reduces unknown unicast flooding.

Route Type 3: Inclusive Multicast Ethernet Tag Route

This route establishes the connection for broadcast, unknown unicast, and multicast (BUM) traffic from a source PE to a remote PE. This route is advertised on per VLAN and per ESI basis.

Route Type 4: Ethernet Segment Route

Ethernet segment routes enable to connect a CE device to two or PE devices. ES route enables the discovery of connected PE devices that are connected to the same Ethernet segment.

Configure EVPN Layer 2 Bridging Service

```
Router # configure
Router (config)# l2vpn
Router (config-l2vpn)# bridge group 1
Router (config-l2vpn-bg)# bridge-domain 1-1
Router (config-l2vpn-bg-bd)# interface GigabitEthernet 0/0/0/1.1
Router (config-l2vpn-bg-bd-ac)# evi 1
Router (config-l2vpn-bg-bd-evi)# exit
Router (config-l2vpn-bg-bd)# exit
Router (config-l2vpn-bg)# bridge-domain 1-2
Router (config-l2vpn-bg-bd)# interface gigabitEthernet 0/0/0/1.2
Router (config-l2vpn-bg-bd-ac)# evi 1
Router (config-l2vpn-bg-bd-ac-evi)# exit
```

Running Configuration

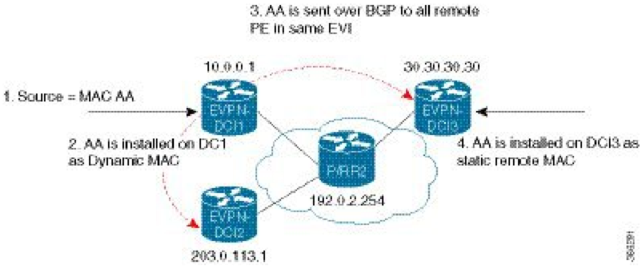
```
l2vpn
 bridge group 1
  bridge-domain 1-1
   interface GigabitEthernet 0/0/0/1.1
    evi 1
    exit
    exit
   bridge-domain 1-2
  interface gigabitEthernet 0/0/0/1.2
 evi 1
```

EVPN Software MAC Learning

MAC learning is the method of learning the MAC addresses of all devices available in a VLAN.

The MAC addresses learned on one device needs to be learned or distributed on the other devices in a VLAN. EVPN Native with Software MAC Learning feature enables the distribution of the MAC addresses learned on one device to the other devices connected to a network. The MAC addresses are learnt from the remote devices using BGP.

Figure 11: EVPN Software MAC Learning



The above figure illustrates the process of software MAC learning. The following are the steps involved in the process:

1. Traffic comes in on one port in the bridge domain.
2. The source MAC address (AA) is learnt on DCI1 and is stored as a dynamic MAC entry.
3. The MAC address (AA) is converted into a type-2 BGP route and is sent over BGP to all the remote PEs in the same EVI.
4. The MAC address (AA) is updated on the DCI3 as a static remote MAC address.

Configure EVPN Native with Software MAC Learning

The following section describes how you can configure EVPN Native with Software MAC Learning:

```
/* Configure bridge domain. */

Router(config)# l2vpn
Router(config-l2vpn)# bridge group EVPN_SH
Router(config-l2vpn-bg)# bridge-domain EVPN_2001
Router(config-l2vpn-bg-bd)# interface TenGigE0/4/0/10.2001
Router(config-l2vpn-bg-bd)# interface BundleEther 20.2001
Router(config-l2vpn-bg-bd)# storm-control broadcast pps 10000
Router(config-l2vpn-bg-bd)# neighbor 20.20.20.20 pw-id 1020001
Router(config-l2vpn-bg-bd-nbr)# evi 2001
Router(config-l2vpn-bg-bd)# exit
Router(config-l2vpn-bg)# exit
Router(config-l2vpn)# exit
```

```
/* Configure advertisement of MAC routes, suppress unknown unicast, disable the control
word,*/
```

```
/* configure the flow label, configure BGP route-exchange using RT. */

Router(config)# evpn
Router(config-evpn)# evi 2001
/*Use the advertise-mac command to control the advertisement of MAC routes through BGP to
other neighbors. */
Router(config-evpn-evi)# advertise-mac
/* Use the unknown-unicast-suppress command to prevent unknown unicast traffic from going
to the MPLS core */
/* and then to all remote PE bridge-ports. */
Router(config-evpn-evi)# unknown-unicast-suppress
/* Use the control-word-disable command to prevent the control word from being sent */
/* in the packet that is sent to MPLS core. The control word functionality is enabled by
default. */
Router(config-evpn-evi)# control-word-disable
/* Perform the following steps to configure BGP route-exchange using RT */
Router(config-evpn-evi)# bgp
Router(config-evpn-evi)# route-target import 200:101
Router(config-evpn-evi)# route-target export 200:101
```

```
/* Configure address family session in BGP. */

Router# configure
Router(config)# router bgp 200
Router(config-bgp)# bgp router-id 40.40.40.40
Router(config-bgp)# address-family l2vpn evpn
Router(config-bgp)# neighbor 10.10.10.10
Router(config-bgp-nbr)# remote-as 200
Router(config-bgp-nbr)# description MPLSFACINGPEER
Router(config-bgp-nbr)# update-source Loopback 0
Router(config-bgp-nbr)# address-family l2vpn evpn
```

Single Home Device or Single Home Network

The following section describes how you can configure EVPN Native with Software MAC Learning feature in single home device or single home network:

In the above figure, the PE (PE1) is attached to Ethernet Segment using bundle or physical interfaces. Null Ethernet Segment Identifier (ESI) is used for SHD/SHN.

Configure EVPN in Single Home Device or Single Home Network

```
/* Configure bridge domain. */

Router(config)# l2vpn
Router(config-l2vpn)# bridge group EVPN_ALL_ACTIVE
Router(config-l2vpn-bg)# bridge-domain EVPN_2001
Router(config-l2vpn-bg-bd)# interface BundleEther1.2001
Router(config-l2vpn-bg-bd)# evi 2001
```

```
/* Configure advertisement of MAC routes. */

Router(config)# evpn
Router(config-evpn)# evi 2001
Router(config-evpn-evi)# advertise-mac
```

```
/* Configure address family session in BGP. */

Router# configure
Router#(config)# router bgp 200
```

```
Router#(config-bgp)# bgp router-id 40.40.40.40
Router#(config-bgp)# address-family l2vpn evpn
Router#(config-bgp)# neighbor 10.10.10.10
Router#(config-bgp-nbr)# remote-as 200
Router#(config-bgp-nbr)# description MPLSFACING-PEER
Router#(config-bgp-nbr)# update-source Loopback 0
Router#(config-bgp-nbr)# address-family l2vpn evpn
```

Running Configuration

```
l2vpn
bridge group EVPN_ALL_ACTIVE
 bridge-domain EVPN_2001
  interface BundleEther1.2001
  evi 2001
!
evpn
 evi 2001
  advertise-mac
!
router bgp 200 bgp
 router-id 40.40.40.40
 address-family l2vpn evpn
 neighbor 10.10.10.10
  remote-as 200 description MPLS-FACING-PEER
  updatesource Loopback0
  addressfamily l2vpn evpn
```

Verification

Verify EVPN in single home devices.

```
Router# show evpn ethernet-segment interface Te0/4/0/10 detail

Ethernet Segment Id    Interface   Nexthops
--------------------   ----------  ----------
N/A        Te0/4/0/10  20.20.20.20
……………
 Topology :
 Operational : SH
 Configured : Single-active (AApS) (default)
```

Verify EVPN Native with Software MAC Learning

Verify the packet drop statistics.

```
Router# show l2vpn bridge-domain bd-name EVPN_2001 details

Bridge group: EVPN_ALL_ACTIVE, bridge-domain: EVPN_2001, id: 1110,
state: up, ShgId: 0, MSTi: 0
 List of EVPNs:
 EVPN, state: up
 evi: 2001
 XC ID 0x80000458
 Statistics:
 packets: received 28907734874 (unicast 9697466652), sent
76882059953
 bytes: received 5550285095808 (unicast 1861913597184), sent
14799781851396
 MAC move: 0
 List of ACs:
 AC: TenGigE0/4/0/10.2001, state is up
 Type VLAN; Num Ranges: 1
...
 Statistics:
 packets: received 0 (multicast 0, broadcast 0, unknown
unicast 0, unicast 0), sent 45573594908
 bytes: received 0 (multicast 0, broadcast 0, unknown unicast
0, unicast 0), sent 8750130222336
 MAC move: 0
 ........
```

Verify the EVPN EVI information with the VPN-ID and MAC address filter.

```
Router# show evpn evi vpn-id 2001 neighbor

Neighbor IP    vpn-id
-----------   --------
20.20.20.20   2001
30.30.30.30   2001
```

Verify the BGP L2VPN EVPN summary.

```
Router# show bgp l2vpn evpn summary
...
Neighbor    Spk   AS     MsgRcvd MsgSent  TblVer    InQ  OutQ  Up/Down  St/PfxRcd
20.20.20.20 0    200     216739  229871   200781341  0    0     3d00h   348032
30.30.30.30 0    200     6462962 4208831  200781341  10   0     2d22h   35750
```

Verify the MAC updates to the L2FIB table in a line card.

```
Router# show l2vpn mac mac all location 0/6/cPU0

Topo ID Producer Next Hop(s)     Mac Address    IP Address
------- -------- -----------     -------------- ----------
1112    0/6/CPU0 Te0/6/0/1.36001 00a3.0001.0001
```

Verify the MAC updates to the L2FIB table in a route switch processor (RSP).

```
Router# show l2vpn mac mac all location 0/6/cPU0

Topo ID  Producer Next Hop(s)     Mac Address    IP Address
-------  -------- -----------     -------------- ----------
1112     0/6/CPU0 Te0/6/0/1.36001 00a3.0001.0001
```

Verify the summary information for the MAC address.

```
Router# show l2vpn forwarding bridge-domain EVPN_ALL_ACTIVE:EVPN_2001 mac-address location
 0/6/CPU0

.....
Mac Address     Type      Learned from/Filtered on   LC learned   Resync Age/Last Change
Mapped to
0000.2001.5555  dynamic   Te0/0/0/2/0.2001           N/A          11 Jan 14:37:22
N/A <-- local dynamic
00bb.2001.0001 dynamic    Te0/0/0/2/0.2001           N/A          11 Jan 14:37:22
N/A
0000.2001.1111 EVPN       BD id: 1110                N/A          N/A
N/A <-- remote static
00a9.2002.0001 EVPN       BD id: 1110                N/A          N/A
N/A
```

Verify the EVPN EVI information with the VPN-ID and MAC address filter.

```
Router# show evpn evi vpn-id 2001 mac

EVI    MAC address     IP address        Nexthop      Label
----   -------------   -----------       -------      ------
2001   00a9.2002.0001  ::                10.10.10.10  34226      <-- Remote MAC
2001   00a9.2002.0001  ::                30.30.30.30  34202

2001   0000.2001.5555  20.1.5.55


Router# show evpn evi vpn-id 2001 mac 00a9.2002.0001 detail

EVI     MAC address      IP address  Nexthop      Label
----    --------------   ----------  -------      -----
2001    00a9.2002.0001   ::          10.10.10.10  34226

2001    00a9.2002.0001   ::          30.30.30.30  34202

 Ethernet Tag : 0
 Multi-paths Resolved : True <--- aliasing to two remote PE with All-Active load balancing

 Static : No
 Local Ethernet Segment : N/A
 Remote Ethernet Segment : 0100.211b.fce5.df00.0b00
 Local Sequence Number : N/A
 Remote Sequence Number : 0
 Local Encapsulation : N/A
 Remote Encapsulation : MPLS
```

Verify the BGP routes associated with EVPN with bridge-domain filter.

```
Router# show bgp l2vpn evpn bridge-domain EVPN_2001 route-type 2

*> [2][0][48][00bb.2001.0001][0]/104
                        0.0.0.0           0 i <------ locally learnt MAC
*>i[2][0][48][00a9.2002.00be][0]/104
     10.10.10.10 100  0 i <----- remotely learnt MAC
* i 30.30.30.30 100 0 i
```

EVPN Multiple Services per Ethernet Segment

EVPN Multiple Services per Ethernet Segment feature allows you to configure multiple services over single Ethernet Segment (ES). Instead of configuring multiple services over multiple ES, you can configure multiple services over a single ES.

You can configure the Native EVPN service on a single Ethernet Bundle.

Both single-active and all-active multihoming modes are supported. However, both single-active and all-active multihoming cannot be configured on a single ES. You can configure either single-active or all-active multihoming mode on a single ES. But, they can coexist.

Configure EVPN Multiple Services per Ethernet Segment

Consider a customer edge (CE) device connected to two provider edge (PE) devices through Ethernet Bundle interface 22001. Configure multiple services on Bundle Ethernet sub-interfaces.

```
Router# configure
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether22001
Router(config-evpn-ac)# ethernet-segment identifier type 0 ff.ff.ff.ff.ff.ff.ff.ff.ee
Router(config-evpn-ac-es)# bgp route-target 2200.0001.0001
Router(config-evpn-ac-es)# exit
Router(config-evpn)# evi 24001
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# route-target import 64:24001
Router(config-evpn-evi-bgp)# route-target export 64:24001
Router(config-evpn-evi-bgp)# exit
Router(config-evpn-evi)# exit
Router(config-evpn)# evi 21006
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# route-target route-target 64:10000
Router(config-evpn-evi-bgp)# exit
Router(config-evpn-evi)# exit
Router(config-evpn)# evi 22101
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# route-target import 64:22101
Router(config-evpn-evi-bgp)# route-target export 64:22101
Router(config-evpn-evi-bgp)# exit
Router(config-evpn-evi)# exit
Router(config-evpn)# evi 22021
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# route-target import 64: 22021
Router(config-evpn-evi-bgp)# route-target export 64: 22021
Router(config-evpn-evi-bgp)# exit
Router(config-evpn-evi)# exit
Router(config-evpn-evi)# advertise-mac
Router(config-evpn-evi)# exit
Router(config-evpn)# evi 22022
Router(config-evpn-evi)# bgp
Router(config-evpn-evi-bgp)# route-target import 64: 22022
Router(config-evpn-evi-bgp)# route-target export 64: 22022
Router(config-evpn-evi-bgp)# exit
Router(config-evpn-evi)# advertise-mac
Router(config-evpn-evi)# commit
Router(config-evpn-evi)# exit
```

Running Configuration

```
evpn
 interface Bundle-Ether22001
  ethernet-segment   identifier type 0 ff.ff.ff.ff.ff.ff.ff.ff.ee
  bgp route-target 2200.0001.0001
  !
  evi 24001
   bgp
    route-target import 64:24001
    route-target export 64:24001
   !
   evi 21006
    bgp
      route-target 64:100006
   !
    evi 22101
     bgp
       route-target import 64:22101
       route-target export 64:22101
     !
   evi 22021
    bgp
      route-target import 64:22021
      route-target export 64:22021
    !
    advertise-mac
  !
  evi 22022
   bgp
    route-target import 64:22022
    route-target export 64:22022
   !
    advertise-mac
  !
```

EVPN Routing Policy

The EVPN Routing Policy feature provides the route policy support for address-family L2VPN EVPN. This feature adds EVPN route filtering capabilities to the routing policy language (RPL). The filtering is based on various EVPN attributes.

A routing policy instructs the router to inspect routes, filter them, and potentially modify their attributes as they are accepted from a peer, advertised to a peer, or redistributed from one routing protocol to another.

This feature enables you to configure route-policies using EVPN network layer reachability information (NLRI) attributes of EVPN route type 1 to 5 in the route-policy match criteria, which provides more granular definition of route-policy. For example, you can specify a route-policy to be applied to only certain EVPN route-types or any combination of EVPN NLRI attributes. This feature provides flexibility in configuring and deploying solutions by enabling route-policy to filter on EVPN NLRI attributes.

To implement this feature, you need to understand the following concepts:

- Routing Policy Language
- Routing Policy Language Structure
- Routing Policy Language Components
- Routing Policy Language Usage
- Policy Definitions
- Parameterization
- Semantics of Policy Application
- Policy Statements
- Attach Points

For information on these concepts, see Implementing Routing Policy.

Currently, this feature is supported only on BGP neighbor "in" and "out" attach points. The route policy can be applied only on inbound or outbound on a BGP neighbor.

EVPN Route Types

The EVPN network layer reachability information (NLRI) provides different route types.

Table 17: EVPN Route Types

Route Type	Name	Usage
1	Ethernet Auto-Discovery (AD) Route	Few routes sent per ES, carry the list of EVIs that belong to ES
2	MAC/IP Advertisement Route	Advertise MAC, address reachability, advertise IP/MAC binding
3	Inclusive Multicast Ethernet Tag Route	Multicast Tunnel End point discovery
4	Ethernet Segment Route	Redundancy group discovery, DF election

Route Type 1: Ethernet Auto-Discovery (AD) Route

The Ethernet (AD) routes are advertised on per EVI and per ESI basis. These routes are sent per ES. They carry the list of EVIs that belong to the ES. The ESI field is set to zero when a CE is single-homed.

Route Type 2: MAC/IP Advertisement Route

The host's IP and MAC addresses are advertised to the peers within NRLI. The control plane learning of MAC addresses reduces unknown unicast flooding.

Route Type 3: Inclusive Multicast Ethernet Tag Route

This route establishes the connection for broadcast, unknown unicast, and multicast (BUM) traffic from a source PE to a remote PE. This route is advertised on per VLAN and per ESI basis.

Route Type 4: Ethernet Segment Route

Ethernet segment routes enable to connect a CE device to two or PE devices. ES route enables the discovery of connected PE devices that are connected to the same Ethernet segment.

EVPN RPL Attribute

Route Distinguisher

A Route Distinguisher (rd) attribute consists of eight octets. An rd can be specified for each of the EVPN route types. This attribute is not mandatory in route-policy.

Example

```
rd in (1.2.3.4:0)
```

EVPN Route Type

EVPN route type attribute consists of one octet. This specifies the EVPN route type. The EVPN route type attribute is used to identify a specific EVPN NLRI prefix format. It is a net attribute in all EVPN route types.

Example

```
evpn-route-type is 3
```

```
The following are the various EVPN route types that can be used:
1 - ethernet-ad
2 – mac-advertisement
3 - inclusive-multicast
4 - ethernet-segment
5 – ip-advertisement
```

IP Prefix

An IP prefix attribute holds IPv4 or IPv6 prefix match specification, each of which has four parts: an address, a mask length, a minimum matching length, and a maximum matching length. The address is required, but the other three parts are optional. When IP prefix is specified in EVPN route type 2, it represents either a IPv4 or IPv6 host IP Address (/32 or /128). When IP prefix is specified in EVPN route type 5, it represents either IPv4 or IPv6 subnet. It is a net attribute in EVPN route type 2 and 5.

Example

```
destination in (128.47.10.2/32)
destination in (128.47.0.0/16)
destination in (128:47::1/128)
destination in (128:47::0/112)
```

esi

An Ethernet Segment Identifier (ESI) attribute consists of 10 octets. It is a net attribute in EVPN route type 1 and 4, and a path attribute in EVPN route type 2 and 5.

Example

```
esi in (ffff.ffff.ffff.ffff.fff0)
```

etag

An Ethernet tag attribute consists of four octets. An Ethernet tag identifies a particular broadcast domain, for example, a VLAN. An EVPN instance consists of one or more broadcast domains. It is a net attribute in EVPN route type 1, 2, 3 and 5.

Example

```
etag in (10000)
```

mac

The mac attribute consists of six octets. This attribute is a net attribute in EVPN route type 2.

Example

```
mac in (0206.acb1.e806)
```

evpn-originator

The evpn-originator attribute specifies the originating router's IP address (4 or 16 octets). This is a net attribute in EVPN route type 3 and 4.

Example

```
evpn-originator in (1.2.3.4)
```

evpn-gateway

The evpn-gateway attribute specifies the gateway IP address. The gateway IP address is a 32-bit or 128-bit field (IPv4 or IPv6), and encodes an overlay next-hop for the IP prefixes. The gateway IP address field can be zero if it is not used as an overlay next-hop. This is a path attribute in EVPN route type 5.

Example

```
evpn-gateway in (1.2.3.4)
```

EVPN Attributes and Operators

This table summarizes the EVPN attributes and operators per attach points.

Table 18: EVPN Attributes and Operators

Attach Point	Attribute	Match	Attribute-Set
neighbor-in	destination	in	—
	rd	in	—
	evpn-route-type	is	—
	esi	in	Yes
	etag	in	Yes
	mac	in	Yes
	evpn-originator	in	—
	evpn-gateway	in	—
neighbor-out	destination	in	—
	rd	in	—
	evpn-route-type	is	—
	esi	in	Yes
	etag	in	Yes
	mac	in	Yes
	evpn-originator	in	—
	evpn-gateway	in	—

CHAPTER 12

Configuring External Control Plane

This module contains the following topics:

Configuring External Control Plane

In Cisco routers, the route processor (RP) that runs the control plane is hosted in the router chassis along with line cards. Hence, there is a tight coupling of control and data planes. Cisco NCS 6000 supports running up to three SDRs on the same RP. With the external control plane (ECP) feature, two external servers are connected to the NCS 6000 chassis through the control ethernet. This ensures control plane reachability across the NCS 6000 chassis and both the servers and provides the additional system resources that are required to scale beyond the current restriction of three named SDRs.

Table 19: SDR Scale Information

Cisco IOS XR Release	Supported Scale	Supported Dell Server Model
Release 6.4.1	up to 4 named SDRs	Dell R630
Release 6.6.1	up to 6 named SDRs with ISSU	Dell R630
Release 7.0.1	up to 6 named SDRs with ISSU	Dell R640

Note The CLI command `hw-module location <card> bootmedia` is not supported for CC cards. Instead, use integrated Dell Remote Access Controller (iDRAC).

Pre-requisites

This section provides information about the hardware and software requirements for configuring the external control plane feature.

Hardware Requirements for NCS 6000 router

- Two Dell R630 or R640 PowerEdge Servers with enterprise version of integrated Dell Remote Access Controller (iDRAC)
- NCS 6008 Line Card Chassis (2T or 1T). For 2T line cards, ensure that the following hardware components are present:
  - Two FAN Trays (NC6-FANTRAY)
  - Universal Fabric Cards (NC6-FC2-U)
- Two Route Processor cards for NCS6008 (NC6-RP)
- Power Trays with 6 PEM for each Power Tray (NCS-AC-PWRTRAY)

Hardware Requirements for Dell R630 Server

- Dell PERC H730 Integrated RAID Controller with 1GB cache for Dell R630 Server

Hardware Requirements for Dell R640 Server

- Dell PERC H740P Mini (Embedded) with 8GB cache for Dell R640 Server

Pluggagbles common for the pair of Dell Servers

- Four Intel SFP optics for each Dell server (FTLX8571D3BCVIT1) of which two optics for the management connectivity and another two optics for the control ethernet connectivity.

Note Dell servers do not support Cisco Copper or SFP optics.

- Four LC to LC OM3 multimode cables required for the control ethernet connectivity. Two cables for the control Ethernet connectivity between each RP and two server NIC ports (used for Expansion Ethernet).
- Four Cisco SFP+ optics for control Ethernet connectivity (SFP-10G-SR) at the RPs
- Two Cisco 1000BASE-T Copper SFP optics for XR management port connectivity at the RP Sysadmin VM port.
- Four LC to LC OM3 multimode cables required for the management connectivity.

Software Requirements for Dell R630 Server

- Cisco IOS XR 6.4.1 latest version with the required packages.

- Dell iDRAC enterprise edition version 2.41.40.40 or later on the Dell servers.
- Dell server BIOS version 2.4.3 or later.
- Supported browser with latest Java or HTML5 plug-in installed for accessing Dell iDRAC's virtual console.

Software Requirements for Dell R640 Server

- Cisco IOS XR 7.0.1 latest version with the required packages.
- Dell iDRAC enterprise edition version 3.21.26.22 or later on the Dell servers.
- Dell server BIOS version 1.6.12 or later.
- NIC firmware version 18.8.9
- Supported browser with latest Java or HTML5 plug-in installed for accessing Dell iDRAC's virtual console.

Pre-configuration tasks

You should complete the following pre-configuration tasks before configuring the ECP feature:

- Setting up the Connection Between the Router and the Servers , on page 159
- Configuring Dell iDRAC, on page 161
- Setting Up Dell Servers for Enabling ECP, on page 162

Setting up the Connection Between the Router and the Servers

Before configuring ECP, you need to establish control Ethernet connectivity in mesh mode between control Ethernet ports of RPs in NCS6000 and NIC ports of the Dell servers. This step ensures control Ethernet fail over connectivity which ensures minimal loop avoidance protocol (MLAP) availability between the NCS6000 router and Dell servers. MLAP helps to avoid control Ethernet loops. You should use the LC-to-LC OM3 multimode cables to establish the connectivity.

NCS6000 router supports:

- Dell R630 in Cisco IOS XR Release 6.4.x
- Dell R630 in Cisco IOS XR Release 6.6.x
- Dell R640 in Cisco IOS XR Release 7.0.x

Note Do not use a combination of the Dell R630 and the R640 servers. Both servers must be of the same model (either R630 or R640).

The control Ethernet connections between the NCS6000 line card chassis and Dell servers needs to be established as per the following figures:

Figure 12: Control Ethernet and Management Connections for Dell R630

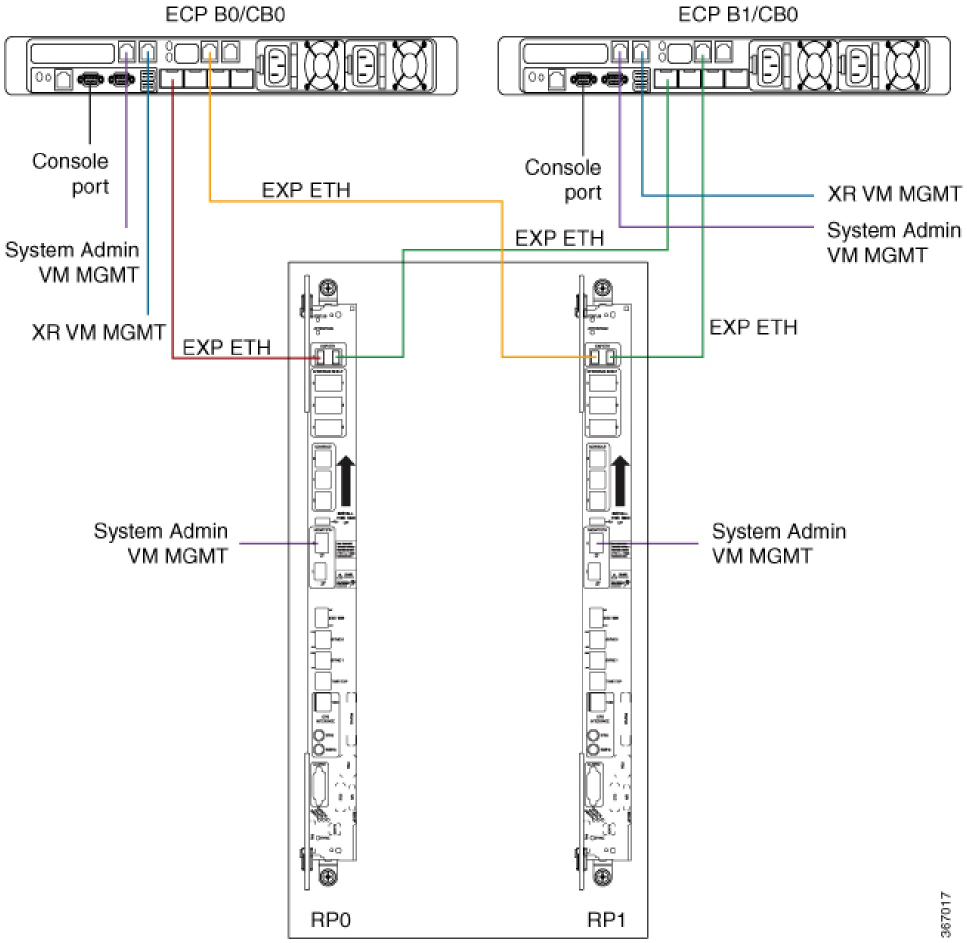
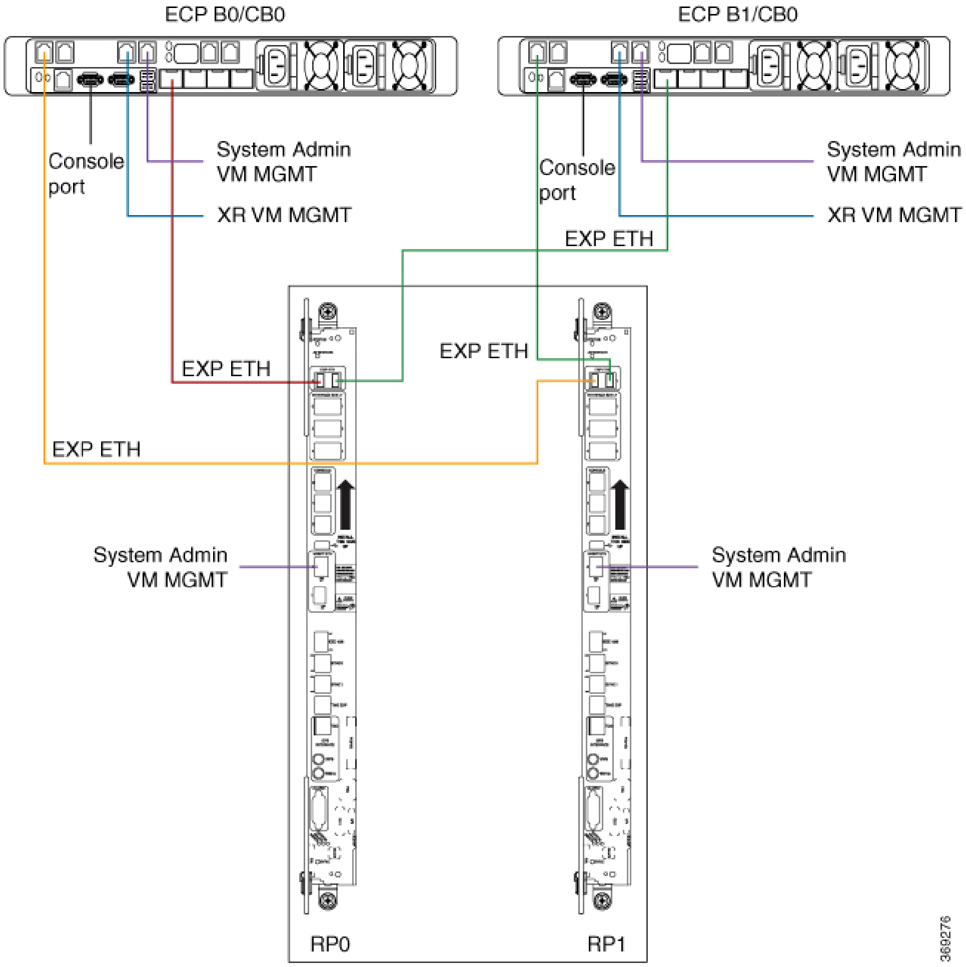


Figure 13: Control Ethernet and Management Connections for Dell R640



Note In Dell R640, the ports connecting to XR VM and System Admin VM are placed in a different location when compared to Dell R630.

You can use one of the following options to connect to the console port on the Dell server:

- RJ-45 to DB-9 Female cable
- RJ-45-to-DB-9 Adapter

Once the Dell servers are powered on, ensure that the power supply LEDs at the cable end of both the AC power supplies display solid green light which indicates that power supplies are working or in good state.

Configuring Dell iDRAC

This section provides information about how to configure the Integrated Dell Remote Access Controller (iDRAC) for Dell servers. The Dell iDRAC allows system administrators to manage the Dell servers remotely.

1. Configure an IP address for iDRAC on the Dell server.
2. Access the iDRAC web interface by specifying the IP address (https://*iDRAC-IP-address*) in the browser address bar.
3. Log in using the default credentials and then create your log in credentials when prompted.
    - Default user name : root
    - Default password : calvin

Note The root user should change the default iDRAC password to ensure secure login to iDRAC.

4. Note down the service tag information for the Dell server from the system summary page. The service tag information is required to update the chassis serial number during the rack configuration.
5. Ensure that iDRAC and BIOS firmware versions on both the servers are same by selecting **System Inventory** > **Firmware Inventory**. Update the firmware versions if required.

Note For more information about firmware update, see http://www.support.dell.com.

Setting Up Dell Servers for Enabling ECP

This section provides information about setting up the Dell servers for enabling ECP. The configuration tasks include:

- Configuring the Dell LifeCycle Controller.
- Using the Dell LifeCycle Controller to configure the following:
    - Serial communication settings in BIOS.
    - Single root I/O virtualization (SR-IOV) mode for NIC ports.
    - Hard disk drive Configuration.
    - NIC Configuration.

Perform the following steps to set up the Dell servers for enabling ECP.

Configuring Dell LifeCycle Controller

Perform the following steps to configure Dell LifeCycle Controller.

1. Launch the iDRAC web interface of the Dell server and log in to iDRAC.
2. In the System Summary, locate **Virtual Console Preview** and click **Launch** to launch the virtual console.
3. In the iDRAC Virtual Console, select **Next Boot** > **LifeCycle Controller** and then click **OK**.

4. Select **Power** > **Power Cycle System (cold boot)** and then click **Yes** in the confirmation window to complete the power cycle.
5. On the LifeCycle Controller page, select the required **Language** and **Keyboard** type and then click **Next**.
6. On the Product Overview page, click **Next**.
7. Enter the network settings on the LifeCycle Controller Network Settings page and then click **Next**.
8. Once the settings are applied, click **Finish**. LifeCycle Controller page is displayed.

Configure the Serial Communication and NIC Port Virtualization Settings Using Dell LifeCycle Controller

Once you configure the Dell Lifecycle Controller, perform the following steps to set the serial communication and NIC port virtualization settings using the Dell LifeCycle Controller. You need to set the virtualization mode as single root I/O virtualization (SR-IOV) mode in both NIC ports.

1. Select **System Setup** and then click **Advanced Hardware Configuration**.
2. In the System BIOS page, click **Serial Communication** and ensure that the following settings are correct.
   - Serial Communication: Auto.
   - Serial Port Address: Serial Device1=COM2, Serial Device2=COM1
   - External Serial Connector: Serial Device1
   - Failsafe Baud Rate: 115200
   - Remote Terminal Type: VT100/VT220
   - Redirection After Boot: Enabled
3. In the System BIOS page, select **Integrated Devices** and ensure that the **SR-IOV Global Enable** setting is **Enabled**.
4. Click **Back** to return to System Setup page.
5. On the Main Configuration page, click **Finish**.
6. Click **Yes** to confirm the changes. The success window appears confirming settings are applied.
7. Click **OK** to return to Device Settings page.

Converting Physical Disks to RAID Capable Mode

Once you complete configuring the NIC port virtualization settings, you need to prepare the disk storage for the NCS 6008 software image installation. First, the physical disk drives need to be converted to RAID capable mode and then virtual hard disks are created. Perform the following steps to convert physical disk drives to RAID capable mode.

Note To configure RAID, a minimum of 4 physical disks are required in the system which can be grouped in to 2 virtual disks.

1. On the Device Settings page, select:

- **Integrated RAID Controller Dell PERC <PERC H730 Mini> Configuration Utility** for Dell R630 server.
- **Integrated RAID Controller Dell PERC <PERC H740 Mini> Configuration Utility** for Dell R640 server.

2. For Dell R630 server:

   - Select **Advanced Controller Management** in the Integrated RAID Controller Dell PERC <PERC H730 Mini> Configuration Utility page

   For Dell R640 server:

   - Select **Advanced Controller Management** in the Integrated RAID Controller Dell PERC <PERC H740 Mini> Configuration Utility page

3. On the Advanced Controller Management page, ensure that the controller mode is RAID.

Note If the controller mode is host bus adapter (HBA), you need to change the controller mode to RAID by selecting **Switch to RAID mode** and then reboot the server.

4. Return to the **Integrated RAID Controller Dell PERC Configuration Utility** page and then select **Configuration Management** .
5. Select **Convert to RAID Capable** to convert the drives as RAID capable. The eligible non-RAID disks are listed.
6. From the list of physical disks, select all the physical disks and then click **OK**.
7. Select **Confirm** and then click **Yes** to confirm the changes.
8. Click **Back** to return to **Configuration Management** .

Creating Virtual Disks

Perform the steps in this procedure to create virtual disks out of the RAID capable disks. In this task, two virtual disks are created. The first RAID 1 virtual disk is created to be hosted on the Solid-State Drive (SSD) units in the Dell servers. A second virtual disk is created with the remaining physical drives so that the first virtual disk appear first while loading the Cisco software.

1. On the Configuration Management page, select **Create Virtual Disk**.
2. Select the RAID level as **RAID 1**, select physical disks from **Unconfigured Capacity**, and then click **Select Physical Disks**.
3. Select the following configuration settings for the physical disks:

   - Select Media Type:

     SSD
   - Select Interface Type:

     Both
   - Logical Sector Size: **Both**

4. Select the two SSD drives and then click **Apply Changes**.

5. Set the virtual disk size as 200 GB and then click **Create Virtual Disk**.

6. Select **Confirm** and then click **Yes** to confirm the changes. The first virtual disk is created.

7. Click **OK** and return to **Configuration Management** to create the second virtual disk.

8. On the Configuration Management page, select **Create Virtual Disk**.

9. Select the RAID level as **RAID 1**, select physical disks from **Free Capacity**, and then click **Select Disk Groups**.

10. Click **Apply Changes** and then **OK** .

11. Use the free space on the disk group for the second virtual disk and then click **Create Virtual Disk**.

12. Select **Confirm** and then click **Yes** to confirm the changes. The second virtual disk is created.

13. Return to **Dashboard View** >**Configuration Management** page.

    In the Dashboard View page, under Properties area the number of Virtual Disks value should be 2.

14. Return to **Device Settings** page.

15. Click **Finish** and return to **System Setup Main Menu** page.
16. Click **Finish** and return to **System Setup** page.
17. Click **Exit** and Virtual Console window opens.

    This completes the storage settings required for the ECP feature configuration on Dell server.

    Repeat the above steps on the second Dell server as well to complete the configuration steps in BIOS and Device Settings page.

Changing the BIOS Boot Sequence

Perform the following steps to change the BIOS boot sequence.

1. Select **System BIOS** > **Boot Settings** > **BIOS Boot Settings**.

2. In BIOS Boot Settings, click **Boot Sequence**.

3. Move **Hard drive C:** to the top of the boot order and then click **OK**.

4. Under **Boot Option Enable/Disable**, uncheck the options except **Hard drive C:** and then click **Back** to return to System BIOS setting.

5. In the System BIOS Settings page, click **Finish** and then click **Yes** to save the changes and click **OK** to the save successful message.

6. On the System Setup page, click **Finish** and then click **Yes** in the confirmation message to reboot the server and the changed BIOS settings to take effect.

Configuring NIC Ports

Perform the following steps to configure NIC ports ettings in the Dell Lifecycle Controller for ECP.

1. Launch the iDRAC web interface of the Dell server and log in to iDRAC.

2. In the System Summary, locate **Virtual Console Preview** and click **Launch** to launch the virtual console.
3. In the iDRAC Virtual Console, select **Next Boot** > **LifeCycle Controller** and then click **OK**.
4. Select **System Settings** and then click **Device Settings**.
5. On the Device setting page, click **Integrated NIC 1 Port 1: Intel ®Ethernet 10G 4P X710 /i350 rNDC**
6. Click **Device Level Configuration** and select the **Virtualization Mode** setting as **SR-IOV**.
7. Return to the **Device Settings** page
8. On the Device Settings page, click **NIC in Slot 1 Port 1: Intel ® Ethernet Converged Network Adapter X710** .
9. Click **Device Level Configuration** and select the **Virtualization Mode** setting as **SR-IOV**.
10. Return to the **Device Settings** page.
11. Repeat the Step 8 and Step 9 for the below two NIC ports participating in the ECP feature:
    - NIC in Slot 2 Port 1: Intel ® Ethernet Converged Network Adapter X710
    - NIC in Slot 2 Port 2: Intel ® Ethernet Converged Network Adapter X710

    This ensures that the Virtualization Mode is set to SR-IOV for all NIC ports relevant to the ECP feature.

Note

- Integrated NIC 1 Port 1: Intel ®Ethernet 10G 4P X710 /i350 rNDC is used for Control Ethernet connectivity
- NIC in Slot 1 Port 1: Intel ® Ethernet Converged Network Adapter X710 is used for Control Ethernet connectivity
- NIC in Slot 2 Port 1: Intel ® Ethernet Converged Network Adapter X710 is used for Management connectivity
- NIC in Slot 2 Port 2: Intel ® Ethernet Converged Network Adapter X710 is used for Management connectivity

Bringing Up the NCS 6008 Router and Dell Servers for Enabling ECP

Perform the steps in this task to bring up the NCS 6008 router chassis and Dell servers for enabling ECP.

Note You need to complete the pre-configuration tasks before performing these steps.

1. Power down both the Dell servers.

2. Ensure that the IOS-XR image running on the chassis supports the ECP feature and any relevant fixes are installed.
3. After the NCS 6008 chassis comes up, ensure that the right version is installed by issuing the **show version** command.
4. Use the **show install active** command to confirm that same version is installed on all nodes.
5. Update the chassis serial numbers of NCS 6008 and Dell servers in the Sysadmin VM running configuration.

```
sysadmin-vm:0_RP0(config)# chassis serial FMP12020050
sysadmin-vm:0_RP0(config-serial-FMP12020050)# rack 0
sysadmin-vm:0_RP0(config)# chassis serial 2YC4JH2
sysadmin-vm:0_RP0(config-serial-2YC4JH2)# rack B0
sysadmin-vm:0_RP0(config)# chassis serial 2YB5JH2
sysadmin-vm:0_RP0(config-serial-2YB5JH2)#  rack B1
sysadmin-vm:0_RP0(config)# commit
```

Note To identify the serial numbers of the remotely managed Dell servers, see Configuring Dell iDRAC, on page 161

6. Power on the Dell servers and load the NCS 6008 image using the virtual media option by performing the following steps:
   a. Launch the Dell iDRAC Virtual Console and then click **Virtual Media** > **Connect Virtual Media**.
   b. Click **Virtual Media** > **Map Removable Disk**.
   c. On the **Virtual Media - Map Removable Disk** dialog box, browse the client system, and select the NCS 6008 bootable IMG image and then click **Map Device**.
   d. In the Virtual Console, click **Next Boot** >**Virtual Floppy** and then click **Ok** to set virtual floppy as the next boot device.
   e. In the Virtual Console, click **Power** > **Power on System** to reboot the server.

      After the server is powered on, NCS 6008 software image installation starts from the mapped software image.

Connecting to Sysadmin VM on the Dell Server through iDRAC SSH

To access terminal console of sysadmin VM running at Dell servers, the root user should establish a SSH connection to the iDRAC IP address from a client running Windows, Linux, or Mac OS which is reachable to the network in which servers are present. Once the SSH connection is established, execute the **console com2** command at the SSH prompt to connect to the server console.

Note After executing the **console com2** command, you may have to press the enter key multiple times to get the redirected console output of the server at the terminal console.

Creating Named SDRs for the ECP Enabled System

Perform the steps in this task to configure a named-SDR and allocate inventory in an ECP enabled system with Dell PowerEdge servers. The inventory includes RP resources (memory and CPU) and line cards. You can repeat the steps to create multiple named-SDRs.

Step 1 **config**

Example:

```
sysadmin-vm:0_RP0# config
```

Enters system administration configuration mode.

Step 2 **sdr** *sdr-name*

Example:

```
sysadmin-vm:0_RP0(config)#  sdr sdr1
```

Creates a named-SDR and Enters SDR configuration mode.

Step 3 **pairing mode inter-rack**

Example:

```
sysadmin-vm:0_RP0(config-sdr-sdr1)#  pairing mode inter-rack
```

Sets the RP pairing to inter-rack mode.

Step 4 **resources card-type cc**

Example:

```
sysadmin-vm:0_RP0(config-sdr-sdr1)#  resources card-type cc
```

Enters RP resources allocation mode for Dell servers. Dell servers are modeled as Compute Container (CC) card in the ECP enabled system.

Step 5 **location** *node-id*

Example:

```
sysadmin-vm:0_RP0(config-sdr-sdr1)#  location B0/CB0
```

Allocates first RP to the named-SDR based on the specified RP location. Here, B0 is the Dell server.

Step 6 **location** *node-id*

Example:

```
sysadmin-vm:0_RP0(config-location-B0/CB0)#  location B1/CB0
```

Allocates second RP to the named-SDR to be used for redundancy. Here B1 is the second Dell server.

Step 7 **exit**

Example:

```
sysadmin-vm:0_RP0(config-location-B1/CB0)#  exit
```

Exits the RP configuration mode and returns to named-SDR configuration mode.

Step 8 **location** *node-id*

Example:

```
sysadmin-vm:0_RP0(config-sdr-sdr1)#  location 0/0
```

Allocates line card to the named-SDR based on the specified line card location.

Step 9 **commit**

Example:

```
sysadmin-vm:0_RP0(config-location-0/0)# commit
```

Example: Creating Named SDRs for ECP

This example shows how to configure named SDRs in a ECP enabled system with external servers. In this example, 4 SDRs are created.

```
sysadmin-vm:0_RP0# config
sysadmin-vm:0_RP0(config)# no sdr default-sdr
sysadmin-vm:0_RP0(config)# commit
sysadmin-vm:0_RP0(config)# sdr sdr1
sysadmin-vm:0_RP0(config-sdr-sdr1)# pairing mode inter-rack
sysadmin-vm:0_RP0(config-sdr-sdr1)# resources card-type cc
sysadmin-vm:0_RP0(config-sdr-sdr1)# location B0/CBO
sysadmin-vm:0_RP0(config-location-B0/CB0)# location B1/CB0
sysadmin-vm:0_RP0(config-location-B1/CB0)# exit
sysadmin-vm:0_RP0(config-sdr-sdr1)# location 0/0
sysadmin-vm:0_RP0(config-location-0/0)# commit

sysadmin-vm:0_RP0(config)# sdr sdr2
sysadmin-vm:0_RP0(config-sdr-sdr2)# pairing mode inter-rack
sysadmin-vm:0_RP0(config-sdr-sdr2)# resources card-type cc
sysadmin-vm:0_RP0(config-sdr-sdr2)# location B0/CB0
sysadmin-vm:0_RP0(config-location-B0/CB0)# location B1/CB0
sysadmin-vm:0_RP0(config-location-B1/CB0)# exit
sysadmin-vm:0_RP0(config-sdr-sdr2)# location 0/3
sysadmin-vm:0_RP0(config-location-0/3)# commit

sysadmin-vm:0_RP0(config)# sdr sdr3
sysadmin-vm:0_RP0(config-sdr-sdr3)# pairing mode inter-rack
sysadmin-vm:0_RP0(config-sdr-sdr3)# resources card-type cc
sysadmin-vm:0_RP0(config-sdr-sdr3)# location B0/CB0
sysadmin-vm:0_RP0(config-location-B0/CB0)# location B1/CB0
sysadmin-vm:0_RP0(config-location-B1/CB0)# exit
sysadmin-vm:0_RP0(config-sdr-sdr3)# location 0/4
sysadmin-vm:0_RP0(config-location-0/4)# commit

sysadmin-vm:0_RP0(config)# sdr sdr4
sysadmin-vm:0_RP0(config-sdr-sdr4)# pairing mode inter-rack
sysadmin-vm:0_RP0(config-sdr-sdr4)# resources card-type cc
sysadmin-vm:0_RP0(config-sdr-sdr4)# location B0/CB0
sysadmin-vm:0_RP0(config-location-B0/CB0)# location B1/CB0
sysadmin-vm:0_RP0(config-location-B1/CB0)# exit
sysadmin-vm:0_RP0(config-sdr-sdr4)# location 0/5
sysadmin-vm:0_RP0(config-location-0/5)# commit


sysadmin-vm:0_RP0(config)# sdr sdr5
sysadmin-vm:0_RP0(config-sdr-sdr5)# pairing mode inter-rack
```

```
sysadmin-vm:0_RP0(config-sdr-sdr5)# resources card-type cc
sysadmin-vm:0_RP0(config-sdr-sdr5)# location B0/CB0
sysadmin-vm:0_RP0(config-location-B0/CB0)# location B1/CB0
sysadmin-vm:0_RP0(config-location-B1/CB0)# exit
sysadmin-vm:0_RP0(config-sdr-sdr5)# location 0/6
sysadmin-vm:0_RP0(config-location-0/6)# commit

sysadmin-vm:0_RP0(config)# sdr sdr6
sysadmin-vm:0_RP0(config-sdr-sdr6)# pairing mode inter-rack
sysadmin-vm:0_RP0(config-sdr-sdr6)# resources card-type cc
sysadmin-vm:0_RP0(config-sdr-sdr6)# location B0/CB0
sysadmin-vm:0_RP0(config-location-B0/CB0)# location B1/CB0
sysadmin-vm:0_RP0(config-location-B1/CB0)# exit
sysadmin-vm:0_RP0(config-sdr-sdr6)# location 0/7
sysadmin-vm:0_RP0(config-location-0/7)# commit
```

What to do next

After the named-SDR are created, verify the VM state for each SDR.

Execute the **show sdr** command to check that the Status is "RUNNING" for all VMs in each SDR.

```
sysadmin-vm:0_RP0# show sdr

Wed Nov  08 16:01:06.626 UTC

SDR: SDR1
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
B0/CB0/VM1   192.0.0.4       RUNNING          1           08/11/2017 00:33:12
B1/CB0/VM1   192.0.4.4       RUNNING          1           08/11/2017 00:33:01
0/1/VM1      192.0.88.3      RUNNING          1           08/11/2017 00:32:48

SDR: SDR2
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
B0/CB0/VM2    192.0.0.6       RUNNING          2           08/11/2017 03:24:43
B1/CB0/VM2    192.0.4.6       RUNNING          2           08/11/2017 03:24:32
0/3/VM2       192.0.68.3      RUNNING          2           08/11/2017 03:25:26

SDR: SDR3
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
B0/CB0/VM3    192.0.0.8       RUNNING          2           08/11/2017 02:32:15
B1/CB0/VM3    192.0.4.8       RUNNING          2           08/11/2017 02:32:23
0/4/VM3       192.0.64.3      RUNNING          2           08/11/2017 02:32:40

SDR: SDR4
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
B0/CB0/VM4    192.0.0.10       RUNNING          2           08/11/2017 04:32:23
B1/CB0/VM4    192.0.4.10       RUNNING          2           08/11/2017 04:32:32
0/5/VM4       192.0.60.3       RUNNING          2           08/11/2017 04:33:40

SDR: SDR5
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
B0/CB0/VM5    192.0.0.12       RUNNING          2           08/11/2017 02:32:17
B1/CB0/VM5    192.0.4.10       RUNNING          2           08/11/2017 02:32:25
0/6/VM5       192.0.66.3       RUNNING          2           08/11/2017 02:32:49

SDR: SDR6
Location     IP Address      Status           Boot Count  Time Started
```

```
-----------------------------------------------------------------------------
B0/CB0/VM6    192.0.0.14       RUNNING          2           08/11/2017 04:32:25
B1/CB0/VM6    192.0.4.12       RUNNING          2           08/11/2017 04:32:37
0/7/VM6       192.0.68.3       RUNNING          2           08/11/2017 04:33:50
```