



System Management Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 26.1.x

First Published: 2026-02-28

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 NON-INFRINGEMENT 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.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE	Preface	ix
----------------	----------------	-----------

CHAPTER 1	New and Changed System Management Features	1
	System Management Features Added or Modified in IOS XR Release 26.x.x	1

CHAPTER 2	YANG Data Models for System Management Features	3
	Using YANG Data Models	3

CHAPTER 3	Configuring Physical and Virtual Terminals	5
	Prerequisites for Implementing Physical and Virtual Terminals	5
	Information About Implementing Physical and Virtual Terminals	5
	Line Templates	5
	Line Template Configuration Mode	6
	Line Template Guidelines	6
	Terminal Identification	7
	vtty Pools	7
	How to Implement Physical and Virtual Terminals on Cisco IOS XR Software	8
	Modifying Templates	8
	Creating and Modifying vtty Pools	9
	Monitoring Terminals and Terminal Sessions	11
	Configuration Examples for Implementing Physical and Virtual Terminals	12

CHAPTER 4	Configuring Simple Network Management Protocol	15
	Prerequisites for Implementing SNMP	15
	Restrictions for SNMP use on Cisco IOS XR Software	15
	Information about Implementing SNMP	16

SNMP Functional Overview	16
SNMP Manager	16
SNMP Agent	16
MIB	16
SNMP Versions	17
Comparison of SNMPv1, v2c, and v3	18
Security Models and Levels for SNMPv1, v2, v3	19
SNMPv3 Benefits	20
SNMPv3 Costs	20
User-Based Security Model	20
View-Based Access Control Model	21
IP Precedence and DSCP Support for SNMP	21
Custom MIB Support Using SNMP Operation Script	22
Restrictions for Custom MIB	23
Create Custom MIB Using SNMP Script	24
Session MIB support on subscriber sessions	25
SNMP Notifications	25
Session Types	26
How to Implement SNMP on Cisco IOS XR Software	27
Configuring SNMPv3	27
Configure to Drop Error PDUs	29
Configuring SNMPv3: Examples	30
Configuring SNMP Trap Notifications	33
Configure to Drop Error PDUs	35
Configuring Trap Notifications: Example	36
Setting the Contact, Location, and Serial Number of the SNMP Agent	37
Defining the Maximum SNMP Agent Packet Size	38
Changing Notification Operation Values	39
Setting IP Precedence and DSCP Values	40
Setting IPv6 Precedence and DSCP Values	40
Setting an IP Precedence Value for SNMP Traffic: Example	41
Setting an IP DSCP Value for SNMP Traffic: Example	42
Displaying SNMP Context Mapping	42
Monitoring Packet Loss	42

Configuring MIB Data to be Persistent	43
Configuring LinkUp and LinkDown Traps for a Subset of Interfaces	44
Polling BRIDGE-MIB	46

CHAPTER 5	Configuring Periodic MIB Data Collection and Transfer	47
	Prerequisites for Periodic MIB Data Collection and Transfer	47
	Information About Periodic MIB Data Collection and Transfer	47
	SNMP Objects and Instances	47
	Bulk Statistics Object Lists	48
	Bulk Statistics Schemas	48
	Bulk Statistics Transfer Options	48
	Benefits of Periodic MIB Data Collection and Transfer	48
	How to Configure Periodic MIB Data Collection and Transfer	49
	Configuring a Bulk Statistics Object List	49
	Configuring a Bulk Statistics Schema	50
	Configuring Bulk Statistics Transfer Options	51
	Periodic MIB Data Collection and Transfer: Example	54

CHAPTER 6	Configuring Cisco Discovery Protocol	57
	Prerequisites for Implementing CDP	57
	Information About Implementing CDP	57
	How to Implement CDP on Cisco IOS XR Software	59
	Enabling CDP	59
	Modifying CDP Default Settings	59
	Monitoring CDP	60
	Examples	61

CHAPTER 7	Configuring Call Home	65
	About Call Home	65
	Benefits of Using Call Home	66
	Prerequisites for Call Home	66
	How to Configure Call Home	67
	Configuring Contact Information	67
	Destination Profiles	69

Configuring and Activating Destination Profiles	69
Call Home Alert Groups	71
Call Home Message Levels	72
Associating an Alert Group with a Destination Profile	73
Configuring Email	76
Configuring a HTTPS Proxy Server	77
Sending Call-home Data through an Email	77
Sending Call-home Data through HTTPS	79
Configuring Call Home to use VRF	81
Configuring Call Home Data Privacy	82
Sending Smart License Data	83

CHAPTER 8

The Network Configuration Protocol	85
Netconf Sessions and Operations	85
The Yang data model	86
Netconf and Yang	87
Supported Yang Models	88
Denial of Services Defense for Netconf-Yang	88
Enabling NETCONF over SSH	89
Examples: Netconf over SSH	90

CHAPTER 9

Configuration and File System Management	93
Secure file transfer from the Router	93
Prerequisites for secure file transfer	95
Secure file transfer using SFTP	95
Secure file transfer using SCP	95
Auto-Save Configuration	96
Configure Auto-Save	98
Auto-Save and Copy Router Configuration Using Public Key Authentication	99
Configuration Example for Auto-Save Using Public Key Authentication	101
Increasing Commit Limit	101
Guidelines and Restrictions for Increasing the Commit Limit	103
Increasing the Rebase Limits	104
Perform ASCII Backup and Rest ASCII Backup Timer	104

Concurrent Configuration Rebase during Commit	105
View VRF-specific Configuration	106
View Filter Running Configuration for VRF	107
Eco Mode Power Saving for fabric and NPU	108
Guidelines for using eco mode and power profiles	109
Configure eco mode power saving	109
Monitor mount points	111
Restrictions for mount-point state monitoring	112

CHAPTER 10**Process Memory Management 113**

Core dump folder limit	113
Restrictions of core dump folder limit	115
How core dump folder limit works	116
Configure core dump folder limit	117



Preface

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



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 26.x.x, on page 1](#)

System Management Features Added or Modified in IOS XR Release 26.x.x

Feature	Description	Changed in Release	Where Documented
Core dump folder limit	This feature is introduced.	Release 25.2.1	Core dump folder limit, on page 113



CHAPTER 2

YANG Data Models for System Management Features

This chapter provides information about the YANG data models for System Management features.

- [Using YANG Data Models, on page 3](#)

Using YANG Data Models

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the [Github](#) repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPath. To view a comprehensive list of the data models supported in a release, navigate to the **Available-Content.md** file in the repository.

You can also view the data model definitions using the [YANG Data Models Navigator](#) tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.



CHAPTER 3

Configuring Physical and Virtual Terminals

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

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

- [Prerequisites for Implementing Physical and Virtual Terminals, on page 5](#)
- [Information About Implementing Physical and Virtual Terminals, on page 5](#)
- [How to Implement Physical and Virtual Terminals on Cisco IOS XR Software, on page 8](#)
- [Configuration Examples for Implementing Physical and Virtual Terminals, on page 12](#)

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 8000 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 8000 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 8000 Series Routers* and *IP Addresses and Services Command Reference for Cisco 8000 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

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
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. <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • end • commit 	Saves configuration changes. <ul style="list-style-type: none"> • When you issue the end command, the system prompts you to commit changes:

	Command or Action	Purpose
	<p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-line)# end</pre> <p>or</p> <pre>RP/0/RP0/CPU0:router(config-line)# commit</pre>	<p>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</p> <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • 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.

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. **vti-pool** {**default** | *pool-name* | **eem**} *first-vty last-vty* [**line-template** {**default** | *template-name*}]
7. Use the **commit** or **end** command.

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	<p>configure</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters global configuration mode.

	Command or Action	Purpose
Step 2	<p>telnet {<i>ipv4</i> <i>ipv6</i>} server max-servers <i>limit</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# telnet ipv4 server max-servers 10</pre>	<p>Specifies the number of allowable Telnet servers. Up to 100 Telnet servers are allowed.</p> <p>Note By default no Telnet servers are allowed. You must configure this command in order to enable the use of Telnet servers.</p>
Step 3	<p>line template <i>template-name</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# line template 1</pre>	<p>Enters line template configuration mode for a user-defined template.</p>
Step 4	<p>Configure the terminal attribute settings for the specified line template using the commands in line template configuration mode.</p>	—
Step 5	<p>exit</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-line)# exit</pre>	<p>Exits line template configuration mode and returns the router to global configuration mode.</p>
Step 6	<p>vtv-pool {default <i>pool-name</i> eem} <i>first-vty last-vty</i> [line-template {default <i>template-name</i>}]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# vty-pool default 0 5 line-template default</pre> <p>or</p> <pre>RP/0/RP0/CPU0:router(config)# vty-pool pool1 5 50 line-template template1</pre> <p>or</p> <pre>RP/0/RP0/CPU0:router(config)# vty-pool eem 100 105 line-template template1</pre>	<p>Creates or modifies vty pools.</p> <ul style="list-style-type: none"> 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. <ul style="list-style-type: none"> 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. <i>pool-name</i> —Creates a user-defined vty pool. <ul style="list-style-type: none"> 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.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • eem —Configures the embedded event manager pool. <ul style="list-style-type: none"> • 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 <i>template-name</i> —Configures the vty pool to reference a user-defined template.
Step 7	Use the commit or end command.	<p>commit —Saves the configuration changes and remains within the configuration session.</p> <p>end —Prompts user to take one of these actions:</p> <ul style="list-style-type: none"> • 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* | **vtty number**]
2. (Optional) **show terminal**
3. (Optional) **show users**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	(Optional) show line [aux location <i>node-id</i> console location <i>node-id</i> vtty number] Example: <pre>RP/0/RP0/CPU0:router# show line</pre>	Displays the terminal parameters of terminal lines. <ul style="list-style-type: none"> • Specifying the show line aux location <i>node-id</i> EXEC command displays the terminal parameters of the auxiliary line.

	Command or Action	Purpose
		<ul style="list-style-type: none"> Specifying the show line console location <i>node-id</i> EXEC command displays the terminal parameters of the console. <ul style="list-style-type: none"> For the location <i>node-id</i> keyword and argument, enter the location of the Route Processor (RP) on which the respective auxiliary or console port resides. The <i>node-id</i> argument is expressed in the format of <i>rack/slot/module</i> . Specifying the show line vty number EXEC command displays the terminal parameters for the specified vty.
Step 2	(Optional) show terminal Example: <pre>RP/0/RP0/CPU0:router# show terminal</pre>	Displays the terminal attribute settings for the current terminal line.
Step 3	(Optional) show users Example: <pre>RP/0/RP0/CPU0:router# show users</pre>	Displays information about the active lines on the router.

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
```



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 tasks you need to implement SNMP on your Cisco IOS XR network.

- [Prerequisites for Implementing SNMP, on page 15](#)
- [Restrictions for SNMP use on Cisco IOS XR Software, on page 15](#)
- [Information about Implementing SNMP, on page 16](#)
- [Custom MIB Support Using SNMP Operation Script, on page 22](#)
- [Session MIB support on subscriber sessions , on page 25](#)
- [How to Implement SNMP on Cisco IOS XR Software, on page 27](#)

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 A 10 Gigabit interface is greater than 2^{32} , so if you are trying to display speed information regarding the interface, you might see concatenated results.

To display correct speed of an interface greater than 10 Gigabit, ifHighSpeed can be used.

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.

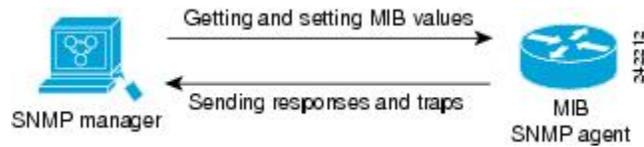
MIB

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.

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 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](#).

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 [Security Models and Levels for SNMPv1, v2, v3, on page 19](#) 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.

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

Table 1: SNMPv1, v2c, and v3 Feature Support

Feature	SNMP v1	SNMP v2c	SNMP v3
Get-Bulk Operation	No	Yes	Yes
Inform Operation	No	Yes	Yes
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 2: SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
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 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 3: 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.

Custom MIB Support Using SNMP Operation Script

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Custom MIB Support Using SNMP Operations Script	Release 25.1.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8011-4G24Y4H-I • 8712-MOD-M
Custom MIB Support Using SNMP Operations Script	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
Custom MIB Support Using SNMP Operations Script	Release 7.5.3	<p>Now you don't have to upgrade to the latest Cisco IOS XR Software release to access a new Management Information Base (MIB). This feature allows you to add a custom script to get support for custom MIB that is not implemented on Cisco IOS XR Software. Custom MIB fetches the required data from an operational database that is already available on the router and returns it on polling the Object Identifier (OID).</p> <p>This feature introduces the following commands:</p> <ul style="list-style-type: none"> • snmp-server script • script snmp <p>This feature also adds the following unified models, you can access these unified models in the Github repository.</p> <ul style="list-style-type: none"> • Cisco-IOS-XR-um-script-cfg • Cisco-IOS-XR-um-script-server-cfg

The MIB is a virtual information storage area for network management information, which consists of collections of managed objects. The OID acts as an identifier to fetch the required data from MIB.

This feature introduces support for custom MIBs that are not implemented in Cisco IOS XR Software. Typically, developing a new MIB is a long and tedious process. Also, you must upgrade to a particular release to get the support of the new MIB.

With this feature, you can define a custom script for a given OID. This custom OID gets the data in the operational database already present on the router and returns it on polling the newly configured OID. SNMP request is sent from Network Management System (NMS) over User Datagram Protocol (UDP) to SNMP daemon. This request spawns customer scripts to fetch data that is related to OID in the request and the output of the script is converted to SNMP protocol data unit and sent to NMS.

Prerequisites

- In the script, the Cython API `snmp_send_response` should be called with data of OID.

Restrictions for Custom MIB

- The length of string data type OIDs must not cross 400 bytes.

Create Custom MIB Using SNMP Script

Configuration Example

In the below example, we create a script which creates OID 1.3.6.1.4.1.9.9.999998.10 to read lldp state.

1. Create a script to fetch required data from the operational database on the router.
2. Use the **describe** command, to fetch the process which executes the command,.

```
Router#describe show lldp
The command is defined in lldp_cmds.parser
```

```
User needs ALL of the following taskids:
```

```
    ethernet-services (READ) or optical (READ)
```

```
It will take the following actions:
```

```
  Spawn the process:
    lldp_command "-s" "-g"
```

The output **lldp_command "-s" "-g"** is used in the following script.

Here is a sample script named **show_lldp_string.py**. This is the command syntax used in the script.

```
import iosxr.snmp
import time
import subprocess as sp
import re
oid = iosxr.snmp.snmp_get_oid()
access_type = iosxr.snmp.snmp_get_access_type()
value = sp.getoutput("lldp_command \'-s\' \'-g\'" )
iosxr.snmp.snmp_send_response("1.3.6.1.4.1.9.9.999998.10", str(value), "OctetString")
```

3. Copy the script file to this location: `harddisk:/mirror/script-mgmt/snmp/`.
4. Use the **sha256sum file-name** command to generate the checksum of the script file.

```
Router:/harddisk:/mirror/script-mgmt/snmp]$sha256sum show_lldp_string.py
```

Here is a sample command output.

```
156345c2cbfc1a2725b5f5ecdfb23d30d9a25e894604890d88929d724946e7b3 show_lldp_string.py
```

5. Enter the configuration mode of the router.
6. Use the **snmp-server community public RW** command to enable read-write community string, where **public** is the read-write community.
7. Use the **snmp-server script script-oid OID-number script-filename file-name** command to map the script file to the custom OID.
8. Use the **script snmp file-name checksum sha256 checksum-value** command to configure the checksum of the script file.

```
Router#configure
```

```
Router(config)#snmp-server community public RW
```

```
Router(config)#snmp-server script script-oid 1.3.6.1.4.1.9.9.999998.10 script-filename
```

```
show_lldp_string.py
```

```
Router(config)#script snmp file-name checksum sha256 checksum-value
```

```
Router(config)#script snmp show_lddp_string.py checksum sha256
156345c2cbfc1a2725b5f5ecdfb23d30d9a25e894604890d88929d724946e7b3
```



- Note**
- The root OID number 1.3.6.1.4.1.9.9.999998 must be used and you can write any Custom OID number after the root OID number.

Yang Data Model for Custom MIB

You can programmatically perform the same configuration using the following unified data models also. To get started with using data models, see the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

Operational Data	Unified Data Model	CLI Commands
Maps script file to the custom OID.	Cisco-IOS-XR-um-script-server-cfg	snmp-server script script-oid <i>OID-number script-filename</i> <i>file-name</i>
Configures checksum for the newly added file-name in the Custom OID.	Cisco-IOS-XR-um-script-cfg	script snmp file-name checksum sha256 checksum-value

Verification

When snmp receives get request for the custom OID, following output is generated:

```
Router # snmpwalk -v2c -c public 5.36.7.100 1.3.6.1.4.1.9.9.999998.10
SNMPv2-SMI::enterprises.9.9.999998.10.0 = STRING: Global LLDP information:
  Status: ACTIVE
  LLDP Chassis ID: 0032.176e.a0df
  LLDP Chassis ID Subtype: MAC Address (IEEE 802-2001) Chassis Subtype
  LLDP System Name: POD-TN3
  LLDP advertisements are sent every 30 seconds
  LLDP hold time advertised is 120 seconds
  LLDP interface reinitialisation delay is 2 seconds
```

Session MIB support on subscriber sessions

SNMP monitoring requires information about subscribers of all types. The CISCO-SUBSCRIBER-SESSION-MIB is defined to model per-subscriber data as well as aggregate subscriber (PPPoE) data. It is required to support notifications (traps) for aggregate session counts crossing configured thresholds. Generic MIB Data Collector Manager (DCM) support for CISCO-SUBSCRIBER-SESSION-MIB, helps faster data collection and also better handling of parallel data.

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.

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

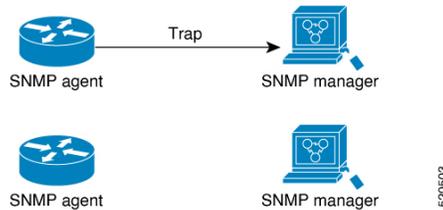
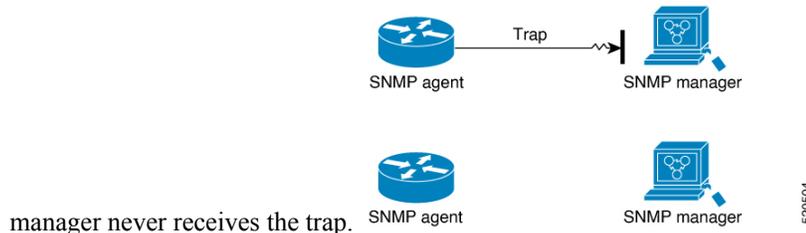


Figure 3: Trap Not Received by the SNMP Manager

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



Note Previously, the SNMP agent would drop traps after 5 minutes if the respective hosts were unreachable. In its new behavior, the SNMP agent now sends traps to all unreachable hosts instead of dropping them. There is no change in the trap delay timer behavior.

Session Types

The supported session types are:

- PPPoE
- IP SUB PKT
- IP SUB DHCP

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 8000 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.

Procedure

Step 1

configure

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2

(Optional) **snmp-server engineid local engine-id**

Example:

```
RP/0/RP0/CPU0:router# snmp-server engineID  
local 00:00:00:09:00:00:00:a1:61:6c:20:61
```

Specifies the identification number of the local SNMP engine.

Step 3

snmp-server view view-name oid-tree {included | excluded}

Example:

```
RP/0/RP0/CPU0:router# snmp-server view  
view_name 1.3.6.1.2.1.1.5 included
```

Creates or modifies a view record.

Step 4 `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# 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 5 `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# snmp-server user
noauthuser group_name v3
```

Configures a new user to an SNMP group.

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`

Example:

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

Displays information about the status of SNMP.

Step 8 (Optional) `show snmp engineid`

Example:

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

Displays information about the local SNMP engine.

Step 9 (Optional) `show snmp group`

Example:

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

Displays information about each SNMP group on the network.

Step 10 (Optional) `show snmp users`

Example:

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

Displays information about each SNMP username in the SNMP users table.

Step 11 (Optional) **show snmp view**

Example:

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

Displays information about the configured views, including the associated MIB view family name, storage type, and status.

Configure to Drop Error PDUs

Perform this configuration to avoid error PDUs being sent out of router when polled with incorrect SNMPv3 user name. If the configuration is not set, it will respond with error PDUs by default. After applying this configuration, when router is polled with unknown SNMPv3 user name, the NMS will get time out instead of getting unknown user name error code.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **snmp-server drop unknown-user**

Example:

```
RP/0/RP0/CPU0:router(config)# snmp-server drop unknown-user
```

Drop the error PDUs when the router is polled with incorrect SNMPv3 user name.

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.
-

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# 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.

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: localSmpID
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# snmp-server user authuser group_name v3 auth md5 clear auth_passwd
```



Note As 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 As 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 SNMP Trap Notifications

The following example shows how to configure the router to send SNMP trap notifications.

Procedure

- Step 1** **configure**
- Example:**
- ```
RP/0/RP0/CPU0:router# configure
```
- Enters global configuration mode.
- Step 2** **snmp-server group name {v1v2v3 {auth | noauth | priv}} [readview] writeview [notifyview] [access-list-name]**
- Example:**
- ```
RP/0/RP0/CPU0:router# snmp-server group group_name v3 noauth read view_name1 writer view_name2
```
- Configures a new SNMP group or a table that maps SNMP users to SNMP views.
- Step 3** **snmp-server user groupname {v1v2cv3 {auth | md5 | sha} {clear | encrypted} auth-password [priv des56 {clear | access-list-name}]**
- Example:**
- ```
RP/0/RP0/CPU0:router# snmp-server group group_name v3 noauth read view_name1 writer view_name2
```
- Configures a new SNMP group or a table that maps SNMP users to SNMP views.
- Step 4** **snmp-server user username groupname {v1v2cv3 {auth | md5 | sha} {clear | encrypted} auth-password [priv des56 {clear | access-list-name}]**
- Example:**
- ```
RP/0/RP0/CPU0:routerconfig# snmp-server user noauthuser group_name v3
```
- Configures a new SNMP group or a table that maps SNMP users to SNMP views.
- Step 5** **[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 6** **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 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.

Step 8 (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.

Configure to Drop Error PDUs

Perform this configuration to avoid error PDUs being sent out of router when polled with incorrect SNMPv3 user name. If the configuration is not set, it will respond with error PDUs by default. After applying this configuration, when router is polled with unknown SNMPv3 user name, the NMS will get time out instead of getting unknown user name error code.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **snmp-server drop unknown-user**

Example:

```
RP/0/RP0/CPU0:router(config)# snmp-server drop unknown-user
```

Drop the error PDUs when the router is polled with incorrect SNMPv3 user name.

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.

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, an authNoPriv user, and an AuthPriv user.



Note The default User Datagram Protocol (UDP) port is 161. If you do not 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 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.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration 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.
- **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.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration 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.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration 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:

- **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 IPv4 Precedence or IPv4 DSCP for SNMP traffic.

Before you begin

SNMP must be configured.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration 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 IPv4 precedence or IPv4 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.
 - **No** —Exits the configuration session without committing the configuration changes.
 - **Cancel** —Remains in the configuration session, without committing the configuration changes.
-

Setting IPv6 Precedence and DSCP Values

This task describes how to configure IPv6 Precedence or IPv6 DSCP for SNMP traffic.

Before you begin

SNMP must be configured.

Procedure**Step 1** **configure****Example:**

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 Use one of the following commands:

- **snmp-server ipv6 precedence** *value*
- **snmp-server ipv6 dscp** *value*

Example:

```
RP/0/RP0/CPU0:router(config)# snmp-server dscp 24
```

Configures an IPv6 precedence or IPv6 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.
- **No** —Exits the configuration session without committing the configuration changes.
- **Cancel** —Remains in the configuration session, without committing the configuration changes.

Setting an IP Precedence Value for SNMP Traffic: Example

The following example shows how to set the SNMP IPv4 Precedence value to 7:

```
configure
 snmp-server ipv4 precedence 7
 exit

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

The following example shows how to set the SNMP IPv6 Precedence value to 7:

```
configure
 snmp-server ipv6 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 IPv4 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

```

The following example shows how to set the IPv6 DSCP value of SNMP traffic to 45:

```

configure
 snmp-server ipv6 dscp 45
 exit

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

```

Displaying SNMP Context Mapping

The SNMP agent serves queries based on SNMP contexts created by the client features. There is a context mapping table. Each entry in the context mapping table includes a context name, the name of the feature that created the context, and the name of the specific instance of the feature.

Procedure

show snmp context-mapping

Example:

```
RP/0/RP0/CPU0:router# show snmp context-mapping
```

Displays the SNMP context mapping table.

Monitoring Packet Loss

It is possible to monitor packet loss by configuring the generation of SNMP traps when packet loss exceeds a specified threshold. The configuration described in this task enables the creation of entries in the MIB tables of the EVENT-MIB. This can then be monitored for packet loss using SNMP GET operations.

Before you begin



Note Entries created in the EVENT-MIB MIB tables using the configuration described in this task cannot be altered using an SNMP SET.

Entries to the EVENT-MIB MIB tables created using an SNMP SET cannot be altered using the configuration described in this task.

Procedure

```
snmp-server mibs eventmib packet-loss type interface-path-id falling lower-threshold interval sampling-interval rising upper-threshold
```

Example:

```
RP/0/RP0/CPU0:router(config)# snmp-server mibs eventmib packet-loss falling 1 interval 5 rising 2
```

Generates SNMP EVENT-MIB traps for the interface when the packet loss exceeds the specified thresholds. Up to 100 interfaces can be monitored.

falling *lower-threshold* —Specifies the lower threshold. When packet loss between two intervals falls below this threshold and an `mteTriggerRising` trap was generated previously, a SNMP `mteTriggerFalling` trap is generated. This trap is not generated until the packet loss exceeds the upper threshold and then falls back below the lower threshold.

interval *sampling-interval* —Specifies how often packet loss statistics are polled. This is a value between 5 and 1440 minutes, in multiples of 5.

rising *upper-threshold* —Specifies the upper threshold. When packet loss between two intervals increases above this threshold, a SNMP `mteTriggreRising` trap is generated. This trap is not generated until the packet loss drops below the lower threshold and then rises above the upper threshold.

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.

Procedure

Step 1 (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.

Step 2 (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 3 (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 4 `snmp-server ifindex persist`

Example:

```
RP/0/RP0/CPU0:router(config)# snmp-server ifindex persist
```

Enables if Index 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.

Procedure

Step 1 `configure`

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

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.

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.

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.

Polling BRIDGE-MIB

BRIDGE-MIB defines the managed objects for MAC-bridges between LAN segments, based on the IEEE802.1d standard. This MIB also supports managing Transparent Bridges, which includes Control-Ethernet and VPLS bridges.

To poll this MIB, do one of the following:

- For SNMPv2: Use a community and map to the context with proper name
- For SNMPv3: Use a group attached to the context

To display the SNMP context mapping table, use the **show snmp context-mapping** command:

```
RP/0/RP0/CPU0:router# show snmp context-mapping
Context-name          Feature-name          Feature
ControlEthernet0_RP0_CPU0_S0  ControlEthernet0_RP0_CPU0_S0  BRIDGEINST
ControlEthernet0_RP1_CPU0_S0  ControlEthernet0_RP1_CPU0_S0  BRIDGEINST

RP/0/RP0/CPU0:router# show running-config snmp-server
snmp-server community cebridge1 RW SystemOwner
snmp-server context ControlEthernet0_RP0_CPU0_S0
snmp-server community-map cebridge1 context ControlEthernet0_RP0_CPU0_S0
```

In the above example, the community name is **cebridge1**, and the context name is **ControlEthernet0_RP0_CPU0_S0**.

The format of the context name is as follows:

- Control-Ethernet bridges – **ControlEthernetrack_slot_module_[S0|S1]**
- VPLS bridges – **vpls_bridge_domain_name**

To configure the recipient of an SNMP notification operation, use the **snmp-server host** command:

```
RP/0/RSP0/CPU0:router(config)# snmp-server host 223.255.254.249 traps version 2c cebridge1
udp-port 1567
```

To enable BRIDGE-MIB trap notifications, use the **snmp-server traps bridgemib** command:

```
RP/0/RSP0/CPU0:router(config)# snmp-server traps bridgemib
```



CHAPTER 5

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.

- [Prerequisites for Periodic MIB Data Collection and Transfer, on page 47](#)
- [Information About Periodic MIB Data Collection and Transfer, on page 47](#)
- [How to Configure Periodic MIB Data Collection and Transfer, on page 49](#)
- [Periodic MIB Data Collection and Transfer: Example, on page 54](#)

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.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration 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 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
```

Adds a MIB object to the bulk statistics object list. Repeat as desired until all objects to be monitored in this list are added.

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.

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.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
Enters global configuration 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 TenGigE 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.

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.

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.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration 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 **Example:**

(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 TenGigE 0/5/0/11/1
RP/0/RP0/CPU0:router(config-bulk-tr)# schema TenGigE/0-CAR
RP/0/RP0/CPU0:router(config-bulk-tr)# schema TenGigE 0/5/0/11/1
```

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(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.

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.
- 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 **commit** *minutes***Example:**

```
RP/0/RP0/CPU0:router(config-bulk-tr)# retain 60
```

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.

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/username/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 6

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.

- [Prerequisites for Implementing CDP, on page 57](#)
- [Information About Implementing CDP, on page 57](#)
- [Enabling CDP, on page 59](#)
- [Modifying CDP Default Settings, on page 59](#)
- [Monitoring CDP, on page 60](#)

Prerequisites for Implementing CDP

To enable CDP, you must install the CDP package on your router.

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. This table summarizes the TLV definitions for CDP advertisements.

Table 5: 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.
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 example explains how to enable CDP globally on the router and then enable CDP on an interface.

```
Router:# configure
Router(config):# cdp
Router(config):# commit

Router:# configure
Router(config):# interface hundredGigE 0/0/0/4
Router(config-if):# cdp
Router(config-if):# 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.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

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.

- In this example, CDP update packets are configured to be sent at a frequency of once every 20 seconds.

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

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.

Procedure

Step 1 `show cdp entry` *{* | entry-name}* [**protocol** | **version**]

Example:

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

Displays detailed information about neighboring devices discovered using CDP.

Step 4 `show cdp traffic` [**location node-id**]

Example:

```
RP/0/RP0/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
NCS5500	Hu0/0/0/4	15	R	NCS-5500	Hu0/0/0/4

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 hundredGigE 0/0/0/4 detail
```

```
-----
Device ID: NCS5500
SysName : NCS5500
Entry address(es):
  IPv4 address: 40.0.0.2
  IPv6 address: 10:10:10:10::1
Platform: cisco NCS-5500, Capabilities: Router
Interface: HundredGigE0/0/0/4
Port ID (outgoing port): HundredGigE0/0/0/4
Holdtime : 13 sec

Version :
7.1.1.112I

advertisement version: 2
Duplex: full
```

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 NCS5500
```

```
-----
Device ID: NCS5500
SysName : NCS5500
Entry address(es):
  IPv4 address: 40.0.0.2
  IPv6 address: 10:10:10:10::1
Platform: cisco NCS-5500, Capabilities: Router
Interface: HundredGigE0/0/0/4
Port ID (outgoing port): HundredGigE0/0/0/4
Holdtime : 11 sec

Version :
7.1.1.112I

advertisement version: 2
Duplex: full
```

The following is sample output for the **show cdp interface** command. In this example, CDP information related to interface 0/0/0/4 is displayed.

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

```
HundredGigE0/0/0/4 is Up
Encapsulation ether
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: 10, Input: 39  
  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: 10, Input: 39  
  Unrecognize Hdr version: 0, File open failed: 0
```




CHAPTER 7

Configuring Call Home

This module describes the configuring of the Call Home feature.

Table 6: Feature History for Configuring Call Home

Release	Modification
Release 7.0.11	Call Home was introduced

This model contains the following topics:

- [About Call Home, on page 65](#)
- [Benefits of Using Call Home, on page 66](#)
- [Prerequisites for Call Home, on page 66](#)
- [How to Configure Call Home, on page 67](#)
- [Configuring Contact Information, on page 67](#)
- [Destination Profiles, on page 69](#)
- [Call Home Alert Groups, on page 71](#)
- [Configuring Email, on page 76](#)
- [Configuring a HTTPS Proxy Server , on page 77](#)
- [Sending Call-home Data through an Email, on page 77](#)
- [Sending Call-home Data through HTTPS, on page 79](#)
- [Configuring Call Home to use VRF, on page 81](#)
- [Configuring Call Home Data Privacy, on page 82](#)
- [Sending Smart License Data , on page 83](#)

About Call Home

Call Home provides an email and HTTPS based notification for critical system policies. A range of message formats are available for compatibility with pager services or XML-based automated parsing applications. You can use this feature to page a network support engineer, or email a Network Operations Center. You can also use Cisco Smart Call Home services to generate a case with the Technical Assistance Center. The Call Home feature can deliver alert messages containing information about diagnostics and environmental faults and events.

The Call Home feature can deliver alerts to multiple recipients, referred to as Call Home destination profiles. Each profile includes configurable message formats and content categories. A predefined destination is

provided for sending alerts to the Cisco TAC, however you also can define your own destination profiles. When you configure Call Home to send messages, the appropriate CLI show command is executed and the command output is attached to the message. Call Home messages are delivered in the following formats:

- Short text format which provides a one or two line description of the fault that is suitable for pagers or printed reports.
- Full text format which provides fully formatted message with detailed information that is suitable for human reading.
- XML machine-readable format that uses Extensible Markup Language (XML) and Adaptive Messaging Language (AML) XML schema definition (XSD). The AML XSD is published on the Cisco.com website at <http://www.cisco.com>. The XML format enables communication with the Cisco Systems Technical Assistance Center.

The Call Home feature is enabled by default. The Cisco TAC-1 profile is created after the device starts. The default Call Home settings that includes destination address, transport methods, alert-group subscriptions, and more are saved in the CiscoTAC-1 profile. To check the default settings, use the **show call-home profile CiscoTAC-1** command.

Benefits of Using Call Home

The Call Home feature offers the following benefits:

- Multiple message-format options:
 - Short Text—Suitable for pagers or printed reports.
 - Plain Text—Full formatted message information suitable for human reading.
 - XML—Matching readable format using Extensible Markup Language (XML) and Adaptive Markup Language (AML) document type definitions (DTDs). The XML format enables communication with the Cisco Smart Call Home server.
- Multiple concurrent message destinations.
- Multiple message categories, including configuration, environmental conditions, inventory, syslog, and crash events.
- Filtering of messages by severity and pattern matching.
- Scheduling of periodic message sending.

Prerequisites for Call Home

How you configure Call Home depends on how you intend to use the feature. Consider the following requirements before you configure Call Home:

- Obtain e-mail, phone, and street address information for the Call Home contact to be configured so that the receiver can determine the origin of messages received.
- Identify the name or IPv4 address of a primary Simple Mail Transfer Protocol (SMTP) server and any backup servers, if using e-mail message delivery.

- Verify IP connectivity from the router to the e-mail server(s) or the destination HTTP server.
- If Cisco Smart Call Home is used, an active service contract covering the device is required to provide full SCH service.

How to Configure Call Home

To configure the sending of Call Home messages, do the following:

1. Assign contact information.
2. Configure and enable one or more destination profiles.
3. Associate one or more alert groups to each profile.
4. Configure the email server options, if using e-mail message delivery.
5. Enable Call Home.

The above tasks are described in detail in the below procedures.



Note Before enabling Call-Home, you must configure the source interface for HTTPS over IPv6. However, for HTTPS over IPv4, Call-Home works without the source interface.

In case of a dual-stack call-home configuration on the device, the IPv4 address is preferred over the IPv6 address. This may result in IPv6 resolution failure. Due to this limitation, the IPv6 device registration with the licensing server may only be done with a single mode, that is, IPv6 only configuration.

Use the **http client source-interface ipv6** command to configure the source interface.

Configuring Contact Information

Each router must include a contact e-mail address. You can optionally include other identifying information for your system installation.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router(config)# call-home
RP/0/RP0/CPU0:router(config-call-home)#
```

Enters call home configuration mode.

Step 3 **contact-email-addr** *email-address*

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# contact-email-addr
user1@cisco.com
```

Configures the customer email address. Enter up to 200 characters in email address format with no spaces.

Step 4 (Optional) **contract-id** *contract-id-string*

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# contract-id
Contract-identifier
```

Configures the contract ID. Enter up to 64 characters. If you include spaces, you must enclose the entry in quotes ("").

Step 5 (Optional) **customer-id** *customer-id-string*

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# customer-id Customer1
```

Configures the customer ID. Enter up to 64 characters. If you include spaces, you must enclose the entry in quotes ("").

Step 6 (Optional) **phone-number** *phone-number-string*

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# phone-number +405-123-4567
```

Configures the customer phone number. The number must begin with a plus (+) prefix, and may contain only dashes (-) and numbers. Enter up to 16 characters.

Step 7 (Optional) **street-address** *street-address*

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# street-address "300 E. Tasman Dr.
San Jose, CA 95134"
```

Configures the customer street address where RMA equipment can be shipped. Enter up to 200 characters. If you include spaces, you must enclose the entry in quotes ("").

Step 8 (Optional) **site-id** *site-id-string*

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# site-id SJ-RouterRoom1
```

Configures the site ID for the system. Enter up to 200 characters. If you include spaces, you must enclose the entry in quotes ("").

Step 9 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 10 **show call-home**

Example:

```
RP/0/RP0/CPU0:router# show call-home
```

Displays information about the system contacts.

Destination Profiles

A destination profile includes the following information:

- One or more alert groups—The group of alerts that trigger a specific Call Home message if the alert occurs.
- One or more e-mail or HTTPS destinations—The list of recipients for the Call Home messages generated by alert groups assigned to this destination profile.
- Message format—The format for the Call Home message (short text, full text, or XML).
- Message severity level—The Call Home severity level that the alert must meet before a Call Home message is sent to all e-mail and HTTPS URL addresses in the destination profile. An alert is not generated if the Call Home severity level of the alert is lower than the message severity level set for the destination profile. The inventory and configuration alert groups do not have concept of severity level. They are generated directly.

You can also configure a destination profile to allow periodic inventory update messages by using the inventory alert group that will send out periodic messages daily, weekly, or monthly.

The following predefined destination profiles are supported:

- CiscoTAC-1—Supports the Cisco-TAC alert group in XML message format.

Configuring and Activating Destination Profiles

You must have at least one activated destination profile for Call Home messages to be sent. The CiscoTAC-1 profile exists by default but is not active.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router(config)# call-home
RP/0/RP0/CPU0:router(config-call-home)#
```

Enters call home configuration mode.

Step 3 **profile *profile-name***

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# profile my_profile
RP/0/RP0/CPU0:router(config-call-home-profile)#
```

Enters call home profile configuration mode to configure a new or existing profile.

Step 4 **destination address email *email-address***

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# destination
address email support_me@cisco.com
```

Configures an email address to which Call Home messages are sent for this profile.

Step 5 **destination message-size-limit *max-size***

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# destination
message-size-limit 1000
```

Configures the maximum size of Call Home messages for this profile. Values can be between 50 and 3145728 characters.

Step 6 **destination preferred-msg-format {*short-text* | *long-text* | *xml*}**

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# destination
preferred-msg-format xml
```

Configures the message format for this profile. The default is xml.

Step 7 **destination transport-method [*email* | *hhttp*]**

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# destination
transport-method email
```

Configures the transport method for this profile.

Step 8 **active**

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# active
```

Activates the destination profile.

Note

At least one destination profile must be active for Call Home messages to be sent.

Step 9 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 10 **show call-home profile** {**all** | *profile-name*}

Example:

```
RP/0/RP0/CPU0:router# show call-home profile all
```

Displays information about the destination profile.

Call Home Alert Groups

An alert group is a predefined subset of alerts or events that Call Home detects and reports to one or more destinations. Alert groups allow you to select the set of alerts that you want to send to a predefined or custom destination profile. Alerts are sent to e-mail destinations in a destination profile only if that alert belongs to one of the alert groups associated with that destination profile and if the alert has a Call Home message severity at or above the message severity set in the destination profile.

The following table lists supported alert groups and the default CLI command output included in Call Home messages generated for the alert group.

Table 7: Alert Groups and Executed Commands

Alert Group	Description	Executed Commands
Environmental	Events related to power, fan, and environment-sensing elements such as temperature alarms.	<ul style="list-style-type: none"> • show environment • show logging • show inventory • show environment trace • show diag
Inventory	Inventory status that is provided whenever a unit is cold booted, or when FRUs are inserted or removed. This alert is considered a noncritical event, and the information is used for status and entitlement.	<ul style="list-style-type: none"> • show platform • show version • show diag • show inventory oid
Syslog	Events generated by specific interesting syslog messages	<ul style="list-style-type: none"> • show version • show logging • show inventory
Configuration	User-generated request for configuration or configuration change event.	<ul style="list-style-type: none"> • show version • show running config all • show inventory • show configuration history last 30 • show configuration commit changes last 1
Snapshot	This alert group can be configured for periodic notifications	By default, this alert group has no commands to be run. You can add the required commands that need to be run.

Call Home maps the syslog severity level to the corresponding Call Home severity level for syslog port group messages.

Call Home Message Levels

Call Home allows you to filter messages based on their level of urgency. You can associate each destination profile (predefined and user-defined) with a Call Home message level threshold. The Call Home message level ranges from 0 (lowest level of urgency) to 9 (highest level of urgency). Call Home messages are generated

if they have a severity level equal to or greater than the Call Home message level threshold for the destination profile.

Call Home messages that are sent for syslog alert groups have the syslog severity level mapped to the Call Home message level.



Note Call Home does not change the syslog message level in the message text.

The following table lists each Call Home message level keyword and the corresponding syslog level for the syslog port alert group.

Table 8: Severity and syslog Level Mapping

Call Home Level	Keyword	syslog Level	Description
9	Catastrophic	Not-Applicable	Network-wide catastrophic failure.
8	Disaster	Not-Applicable	Significant network impact.
7	Fatal	Emergency (0)	System is unusable.
6	Critical	Alert (1)	Critical conditions that indicate that immediate attention is needed.
5	Major	Critical (2)	Major conditions.
4	Minor	Error (3)	Minor conditions.
3	Warning	Warning (4)	Warning conditions.
2	Notification	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	Normal	Information (6)	Normal event signifying return to normal state.
	Debugging	Debug (7)	Debugging messages.

Associating an Alert Group with a Destination Profile

An alert is sent only to destination profiles that have subscribed to the Call Home alert group.

Before you begin

Use the **show call-home alert-group** command to view available alert groups.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router(config)# call-home
RP/0/RP0/CPU0:router(config-call-home)#
```

Enters call home configuration mode.

Step 3 **profile *profile-name***

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# profile my_profile
RP/0/RP0/CPU0:router(config-call-home-profile)#
```

Enters call home profile configuration mode to configure a new or existing profile.

Step 4 **subscribe-to-alert-group inventory [periodic {daily | monthly *day-of-month* | weekly *day-of-week*} hh:mm**

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group
inventory periodic monthly 1 10:00
```

Configures a destination profile to receive messages for the inventory alert group. Either alerts are sent periodically, or any non-normal event triggers an alert.

Step 5 **subscribe-to-alert-group syslog severity *severity-level* pattern *string***

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group
syslog severity major pattern
```

Configures a destination profile to receive messages for the syslog alert group. Alerts with a severity the same or greater than the specified severity level are sent.

- **catastrophic**—Includes network-wide catastrophic events in the alert. This is the highest severity.
- **critical**—Includes events requiring immediate attention (system log level 1).
- **disaster**—Includes events with significant network impact.
- **fatal**—Includes events where the system is unusable (system log level 0).
- **major**—Includes events classified as major conditions (system log level 2).

- **minor**—Includes events classified as minor conditions (system log level 3)
- **normal**—Specifies the normal state and includes events classified as informational (system log level 6). This is the default.
- **notification**—Includes events informational message events (system log level 5).
- **warning**—Includes events classified as warning conditions (system log level 4).

You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").

Step 6 **subscribe-to-alert-group snapshot severity *severity-level* pattern *string***

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group
snapshot severity major pattern
```

Configures a destination profile to receive messages for the snapshot alert group. Alerts with a severity the same or greater than the specified severity level are sent.

You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").

Step 7 **subscribe-to-alert-group configuration severity *severity-level* pattern *string***

Example:

```
RP/0/RP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group configuration severity major
pattern
```

Configures a destination profile to receive messages for the configuration alert group. Alerts with a severity the same or greater than the specified severity level are sent.

You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").

Step 8 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

Use the **show call-home profile** command to view the profile configurations.

Configuring Email

If Call Home messages are sent via email, the you must configure your email server before Call Home messages can be sent.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router(config)# call-home  
RP/0/RP0/CPU0:router(config-call-home)#
```

Enters call home configuration mode.

Step 3 (Optional) **sender from** *email-address*

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# sender from  
my_email@cisco.com
```

Specifies the email message “from” address.

Step 4 (Optional) **sender reply-to** *email-address*

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# sender reply-to  
my_email@cisco.com
```

Specifies the email message “reply-to” address.

Step 5 Required: **mail-server address priority priority**

Example:

```
RP/0/RP0/CPU0:router(config-call-home)# mail-server  
198.61.170.16 priority 1
```

Specifies the mail server to use to send Call Home messages. You can specify an IP address or mail server name. You can specify up to five mail servers to use. The server with the lower priority is tried first.

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 **show call-home mail-server status**

Example:

```
RP/0/RP0/CPU0:router# show call-home mail-server status
```

Displays the status of the specified mail server.

Configuring a HTTPS Proxy Server

This task enables the user to configure a HTTPS Proxy Server.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router (config) # call-home
```

Enters Call Home configuration mode.

Step 3 **http-proxy *proxy-server-name* port *port-number***

Example:

```
RP/0/RP0/CPU0:router (config) # http-proxy pl port 100
```

Configures the port for the specified HTTPS proxy server. Range is 1 to 65535.

Sending Call-home Data through an Email

This task enables the user to configure sending Call-home data using email as the transport method:

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router (config) # call-home
```

Enters Call Home configuration mode.

Step 3 **profile *name***

Example:

```
RP/0/RP0/CPU0:router (config-call-home) # profile user1
```

Enters call home destination profile configuration mode for the specified destination profile name. If the specified destination profile does not exist, it is created.

Step 4 **active**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # active
```

Enables the destination profile. By default, a user-defined profile is enabled when it is created.

Step 5 **destination transport-method email**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destination transport-method email
```

Configures the message transport method for email. This is the default

Step 6 **destination address email *email-address***

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destination address email xyz@cisco.com
```

Configures the destination e-mail address to which Call Home messages are sent.

Step 7 **destination preferred-msg-format {*long-text* |*short-text*| xml}**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destinationpreferred-msg-format xml
```

(Optional) Configures a preferred message format. The default is XML.

Step 8 **subscribe-to-alert-group syslog severity *severity-level* pattern *string***

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # subscribe-to-alert-group syslog severity normal
pattern COUNT
```

Configures a destination profile to receive messages for the syslog alert group. Alerts with a severity the same or greater than the specified severity level are sent.

- **critical**—Includes events requiring immediate attention (system log level 1).
- **disaster**—Includes events with significant network impact.
- **fatal**—Includes events where the system is unusable (system log level 0).
- **major**—Includes events classified as major conditions (system log level 2).
- **minor**—Includes events classified as minor conditions (system log level 3).
- **normal**—Specifies the normal state and includes events classified as informational (system log level 6).
This is the default.
- **notification**—Includes events informational message events (system log level 5).
- **warning**—Includes events classified as warning conditions (system log level 4).

You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").

Step 9 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.

Sending Call-home Data through HTTPS

This task enables the user to configure sending Call-home data using HTTPS as the transport method:



Note

For the HTTPS function to work you should use the **crypto ca trustpoint** command to declare a CA, followed by the **crl option** command. This ensures that the certificates of other peers are accepted without trying to obtain the appropriate CRL. For example:

```
RP/0/RP0/CPU0:ios(config)#crypto ca trustpoint Trustpool
RP/0/RP0/CPU0:ios(config-trustp)#crl optional
```

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router (config) # call-home
```

Enters Call Home configuration mode.

Step 3 **profile name**

Example:

```
RP/0/RP0/CPU0:router (config-call-home) # profile user1
```

Enters call home destination profile configuration mode for the specified destination profile name. If the specified destination profile does not exist, it is created.

Step 4 **active**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # active
```

Enables the destination profile. By default, a user-defined profile is enabled when it is created.

Step 5 **destination transport-method http**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destination transport-method http
```

Configures the message transport method for HTTPS.

Step 6 **destination address http url**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destination address http https://example.com
```

Configures the destination URL address to which Call Home messages are sent.

Step 7 **destination preferred-msg-format {long-text |short-text| xml}**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destinationpreferred-msg-format xml
```

(Optional) Configures a preferred message format. The default is XML.

Step 8 **subscribe-to-alert-group syslog severity severity-level pattern string**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # subscribe-to-alert-group syslog severity normal
pattern COUNT
```

Configures a destination profile to receive messages for the syslog alert group. Alerts with a severity the same or greater than the specified severity level are sent.

- **critical**—Includes events requiring immediate attention (system log level 1).
- **disaster**—Includes events with significant network impact.
- **fatal**—Includes events where the system is unusable (system log level 0).
- **major**—Includes events classified as major conditions (system log level 2).
- **minor**—Includes events classified as minor conditions (system log level 3).
- **normal**—Specifies the normal state and includes events classified as informational (system log level 6).
This is the default.
- **notification**—Includes events informational message events (system log level 5).
- **warning**—Includes events classified as warning conditions (system log level 4).

You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").

Step 9 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.

Configuring Call Home to use VRF

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router (config) # call-home
```

Enters Call Home configuration mode.

Step 3 `vrf vrf-name`

Example:

```
RP/0/RP0/CPU0:router (config-call-home) # vrf v1
```

Configures call home for the specified VRF. VRF works only for the http transport method. It does not work for the email transport method.

Configuring Call Home Data Privacy

Procedure

Step 1 `configure`

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 `call-home`

Example:

```
RP/0/RP0/CPU0:router (config) # call-home
```

Enters the call home configuration submode.

Step 3 `data-privacy { level { normal | high } | hostname }`

Example:

```
RP/0/RP0/CPU0:router (config-call-home) # data-privacy level high
```

Scrubs data from call-home message to protect the privacy of the user. The default data-privacy level is normal.

- **normal** - scrubs all normal level commands , such as , password/ username/ ip/ destination.
- **high** - scrubs all normal level commands plus the IP domain name and IP address commands.
- **hostname** - scrubs all high-level or normal-level commands plus the hostname command. It may cause Smart Call Home processing failure.

Note

Enabling the data-privacy command can affect CPU utilization when scrubbing a large amount of data.

Sending Smart License Data

This task enables the user to configure sending Smart License data through HTTPS transport method in TAC or user-defined profile:

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **call-home**

Example:

```
RP/0/RP0/CPU0:router (config) # call-home
```

Enters Call Home configuration mode.

Step 3 **profile name**

Perform either one of the below actions:

- For sending Smart License data in TAC profile:

```
RP/0/RP0/CPU0:router (config-call-home) # profile CiscoTAC-1
```

- For sending Smart License data in user-defined profile:

```
RP/0/RP0/CPU0:router (config-call-home) # profile user1
```

Step 4 **active**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # active
```

Enables the destination profile. By default, a user-defined profile is enabled when it is created.

Step 5 **reporting smart-licensing-data**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # reporting smart-licensing-data
```

Enables sending Smart Licensing data.

Step 6 **destination transport-method http**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destination transport-method http
```

Configures the message transport method for HTTPS.

Step 7 **destination address http url**

Example:

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destination address http https://example.com
```

Configures the destination HTTPS address to which Smart License data is sent.

Step 8 destination preferred-msg-format {long-text |short-text| xml}**Example:**

```
RP/0/RP0/CPU0:router (config-call-home-profile) # destinationpreferred-msg-format xml
```

(Optional) Configures a preferred message format. The default is XML.

Step 9 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.
-



CHAPTER 8

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, as defined in RFC6241. Yang is a data modeling language used with Netconf, as defined in RFC6020.

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.

Netconf runs within a Secure Shell (SSH) session as an SSH subsystem, as defined in RFC6242.

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.

- [Netconf Sessions and Operations, on page 85](#)
- [The Yang data model , on page 86](#)
- [Netconf and Yang, on page 87](#)
- [Supported Yang Models , on page 88](#)
- [Denial of Services Defense for Netconf-Yang, on page 88](#)
- [Enabling NETCONF over SSH, on page 89](#)

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
(exec-19.42.10) bash-4.2$ pyang -f tree Cisco-IOS-XR-aaa-lib-cfg.yang
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? dt1:Aaa-accounting-rp-failover
      |   +--rw broadcast?  dt1:Aaa-accounting-broadcast
      |   +--rw type-xr?    dt1:Aaa-accounting
      |   +--rw method1?   dt1:Aaa-method-accounting
      |   +--rw method2?   dt1:Aaa-method-accounting
      |   +--rw method3?   dt1:Aaa-method-accounting
      |   +--rw method4?   dt1:Aaa-method-accounting
      |   +--rw server-group-name1? string
      |   +--rw server-group-name2? string
      |   +--rw server-group-name3? string
      |   +--rw server-group-name4? string
    +--rw authorizations
      | +--rw authorization* [type listname]
      |   +--rw type          xr:Cisco-ios-xr-string
      |   +--rw listname     xr:Cisco-ios-xr-string
      |   +--rw method1?    dt1:Aaa-method
      |   +--rw method2?    dt1:Aaa-method
      |   +--rw method3?    dt1:Aaa-method
      |   +--rw method4?    dt1:Aaa-method
      |   +--rw server-group-name1? string
      |   +--rw server-group-name2? string
      |   +--rw server-group-name3? string
      |   +--rw server-group-name4? string
    +--rw accounting-update!
      | +--rw type          dt1:Aaa-accounting-update
      | +--rw periodic-interval? uint32
```

```

+--rw banner
| +--rw login?  string
+--rw authentications
  +--rw authentication* [type listname]
    +--rw type                xr:Cisco-ios-xr-string
    +--rw listname            xr:Cisco-ios-xr-string
    +--rw method1?           dt1:Aaa-method
    +--rw method2?           dt1:Aaa-method
    +--rw method3?           dt1:Aaa-method
    +--rw method4?           dt1:Aaa-method
    +--rw server-group-name1? string
    +--rw server-group-name2? string
    +--rw server-group-name3? string
    +--rw server-group-name4? 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 8000 Series Router with Netconf capability
- Netconf Client Application with connection to the router

S. No.	Device / component	Action
1	Cisco router	Login/ access the router.
2	Cisco router	Prerequisites for enabling Netconf: <ul style="list-style-type: none"> • Crypto keys must be generated.
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.

S. No.	Device / component	Action
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 88
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 Defense 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 8000 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.

Prerequisite:

- Crypto keys must be generated prior to this configuration.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

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 subsystem 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 used 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 0ms|
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|

```




CHAPTER 9

Configuration and File System Management

This module describes methods for configuration management and file transfer enhancements.

- [Secure file transfer from the Router, on page 93](#)
- [Auto-Save Configuration, on page 96](#)
- [Auto-Save and Copy Router Configuration Using Public Key Authentication, on page 99](#)
- [Increasing Commit Limit, on page 101](#)
- [View VRF-specific Configuration, on page 106](#)
- [Eco Mode Power Saving for fabric and NPU, on page 108](#)
- [Monitor mount points, on page 111](#)

Secure file transfer from the Router

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
Secure file transfer from the Router	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: <ul style="list-style-type: none"> • 8011-12G12X4Y-A • 8011-12G12X4Y-D
Secure file transfer from the Router	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*) *This feature is supported on: <ul style="list-style-type: none"> • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
Secure file transfer from the Router	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Secure file transfer from the Router	Release 7.9.1	<p>Your routers are now enabled to transfer files securely to an archive server. It's made possible because the copy command now supports SFTP (Secure File Transfer Protocol) and SCP (Secure Copy Protocol) using the underlying SSH protocol implementation. Secure transfer of files from the router maintains the integrity, confidentiality, and availability of network configurations.</p> <p>This feature modifies the copy command.</p>

You can duplicate files or data in the router from one location to another using the **copy** command. This functionality helps to create a copy of a file, folder, or data set and place it in a specific destination. You can use the copy functionality to back up files, move data between directories, create duplicates of the files for editing or distribution without modifying the original content. It also allows you to retain the original data while making a duplicate that you can further manipulate independently.

Starting with Cisco IOS XR Release 7.9.1, we've enhanced the functionality of the copy command to support secure file transfer from the router. Secure file transfer protects data during transit using the SFTP (Secure File Transfer Protocol) and SCP (Secure Copy Protocol) when sharing files within or across networks. The SFTP and SCP functionalities in the copy feature use the SSH protocol implementation in the router to secure transfer the files to a remote server.

You can use the following options in the **copy** command for secure file transfer:

- **sftp**: You can transfer the files to a remote location using the **SFTP** file transfer protocol. SFTP is a secure file transfer protocol for transferring large files.
- **scp**: You can transfer the files to a remote location using the **SCP** file transfer protocol. SCP is a secure copy protocol to transfer files between servers.

Starting Cisco IOS XR Software Release 7.10.1, you can use public-key authentication while copying the running configuration.

Configuration Example for SCP and SFTP Using Public-Key Authentication

While you're using public-key authentication for copying running configuration from the router to a remote server, you don't need to mention **password** in the command. The following example shows how you can configure public-key authentication while copying configuration using the SCP protocol:

```
Router#copy running-config scp://root@192.0.4.2//var/opt/run_conf_scp.txt
```

Prerequisites for secure file transfer

Enable the SSH Server in the router:

```
Router# config
Router(config)# ssh server v2
Router(config)# ssh server vrf default
Router(config)# ssh server netconf vrf default
Router(config)# commit
```

Secure file transfer using SFTP

You can copy the running configuration file from the router to a remote server using SFTP as follows:

```
Router# copy running-config sftp://root:testpassword@192.0.2.1//var/opt/run_conf_sftp.txt
```

```
Destination file name (control-c to cancel): [/var/opt/run_conf_sftp.txt]?
```

```
.
215 lines built in 1 second
[OK]Connecting to 192.0.2.1...22
Password:
sftp> put /tmp/tmpsymlink/nvgen-34606-_proc_34606_fd_75 /var/opt/run_conf_sftp.txt
```

```
/tmp/tmpsymlink/nvgen-34606-_proc_34606_fd_75
```

```
Transferred 3271 Bytes
3271 bytes copied in 0 sec (3271000)bytes/sec
sftp> exit
```

Verification in the SFTP Server

```
[root@sftp_server ~]# ls -ltr /var/opt/run_conf_sftp.txt
-rw-r--r-- 1 root root 3271 Mar 21 18:07 /var/opt/run_conf_sftp.txt
```

Secure file transfer using SCP

You can copy the running configuration file from the router to a remote server using SFTP as follows:

```
Router# copy running-config sftp://root:testpassword@192.0.2.1//var/opt/run_conf_sftp.txt
```

```
Destination file name (control-c to cancel): [/var/opt/run_conf_sftp.txt]?
```

```
.
215 lines built in 1 second
[OK]Connecting to 192.0.2.1...22
Password:
```

```
sftp> put /tmp/tmpsymlink/nvgen-34606-_proc_34606_fd_75 /var/opt/run_conf_sftp.txt

/tmp/tmpsymlink/nvgen-34606-_proc_34606_fd_75

Transferred 3271 Bytes
3271 bytes copied in 0 sec (3271000)bytes/sec
sftp> exit
```

Verification in the SFTP Server

```
[root@sftp_server ~]# ls -ltr /var/opt/run_conf_sftp.txt
-rw-r--r-- 1 root root 3271 Mar 21 18:07 /var/opt/run_conf_sftp.txt
```

Auto-Save Configuration

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Auto-Save with Secure File-Transfer and Additional Configurable Parameters	Release 25.1.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8011-4G24Y4H-I • 8712-MOD-M
Auto-Save with Secure File-Transfer and Additional Configurable Parameters	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
Auto-Save with Secure File-Transfer and Additional Configurable Parameters	Release 7.9.1	<p>Apart from automatically backing up the running configuration after every commit, you can also do the following with Auto-Save:</p> <ul style="list-style-type: none"> • Save running configurations to remote systems using Secure Copy Protocol (SCP) and Secure File Transfer Protocol (SFTP). • Configure wait-time between two subsequent auto-saves. • Append time-stamp to the file name of the saved configuration. • Save the encrypted password. • Specify the maximum number of files that you can auto-save. <p>The feature introduces these changes:</p> <p>CLI: Modified the configuration commit auto-save command by adding the following keywords:</p> <ul style="list-style-type: none"> • filename scp • filename sftp • wait-time • timestamp • password • maximum <p>Yang Data Model:</p> <ul style="list-style-type: none"> • New XPath for Cisco-IOS-XR-config-autosave-cfg • New XPath for Cisco-IOS-XR-um-config-commit-cfg

You can configure the router to automatically take the backup of the running configuration by using **configuration commit auto-save** command. This auto-save feature saves the configuration to the specified location on the router after every **commit** is made. These auto-save files are stored in the form of Linux files.

Starting Cisco IOS XR Software Release 7.9.1, the auto-save feature is enhanced to provide a set of functionalities. Use the following keywords to achieve the same:

- **scp and sftp** - You can save the running configuration backup files to remote location using **scp** and **sftp** file transfer protocols. SCP is a secure copy protocol to transfer files between servers. Whereas SFTP is a secure file transfer protocol for transferring large files.
- **password** - You can save encrypted passwords for the remote and non-remote URLs.
- **maximum** - You can mention maximum number of files that can be saved automatically. Once the maximum number of auto-saved file is reached, the newer auto-save files starts replacing the older auto-save files. The default value of **maximum** is 1. You can save upto 4294967295 files.
- **timestamp** - Using this keyword, the time-stamp can be appended to the auto-saved configuration file name. The **timestamp** uses the time and timezone configured on the router. The saved file displays timestamp in <day> <month> <date> <hours> <minutes> <seconds> <milliseconds> format. Here is an example of auto-saved file with time-stamp - : *test_123.autosave.1.ts.Tue_Jan_31_15-15-51_805_IST*
- **wait-time** - You can specify how long to wait before next auto-save happens in terms of days, months or hours after the commit is made. The default value of **wait-time** is zero.

Restriction for Auto-Save Configuration

The auto-save configuration is only available on the local paths, scp, and sftp paths.

Starting Cisco IOS XR Software Release 7.10.1, you can use public-key authentication while automatically saving the running configuration. For more detailed information on how to use public-key authentication, see [Auto-Save and Copy Router Configuration Using Public Key Authentication, on page 99](#).



Note The maximum number of configuration lines currently supported on the Cisco 8000 routers is 1M.

Configure Auto-Save

Use the **configuration commit auto-save** command to auto save the configuration.

```
Router#configure
Router(config)#configuration commit auto-save
Router(config-cfg-autosave)#commit
```

You can also configure options such as **password**, **timestamp**, **maximum**, and **wait-time** with the **configuration commit auto-save** command. The location to save the file-name must be specified in <protocol>://<user>@<host>:<port>/<url-path>/<file-name> format.

```
Router(config-cfg-autosave)#configuration commit auto-save filename
sftp://user1@server1://test-folder/test_123
Router(config-cfg-autosave)#password clear encryption-default cisco
Router(config-cfg-autosave)#timestamp
Router(config-cfg-autosave)#maximum 10
Router(config-cfg-autosave)#wait-time days 0 hours 0 minutes 0 seconds 5
Router(config-cfg-autosave)#commit
```

Running Configuration

```
Router#show running-config configuration commit auto-save
configuration commit auto-save
  filename sftp://user1@server1://test-folder/test_123
  password encrypted encryption-default <password for above user>
  timestamp
  maximum 10
  wait-time days 0 hours 0 minutes 0 seconds 5
!
```

Auto-Save and Copy Router Configuration Using Public Key Authentication

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
Auto-Save and Copy Router Configuration Using Public Key Authentication	Release 25.1.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8011-4G24Y4H-I • 8712-MOD-M
Auto-Save and Copy Router Configuration Using Public Key Authentication	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
Auto-Save and Copy Router Configuration Using Public Key Authentication	Release 7.10.1	<p>You can now experience passwordless authentication while automatically saving running configurations and securely copying them on the router. The feature uses public key-based authentication, a secure logging method using a secure shell (SSH), which provides increased data security. This feature offers automatic authentication and single sign-on benefits, which also aids in a secure automation process.</p> <p>The feature introduces these changes:</p> <p>CLI:</p> <ul style="list-style-type: none"> • The configuration commit auto-save command supports password-less authentication. • The copy command supports password-less authentication.

From Cisco IOS XR Software Release 7.10.1, you don't need to remember and enter the **password** as you can use public key-based authentication while doing the following:

- Automatically saving your running configuration
- Copying the configuration from a source (such as a network server) to a destination (such as a flash disk)

Password is automatically verified when you have enabled SSH connection using public key-based authentication. Using public key-based authentication avoids several problems such as password disclosure and password leakage.

Public key is mathematically related to private key. The private key is secret, whereas the public key is available on the servers. You can copy the public key to the SSH server from the SSH client. Then, when you try to secure the running configuration, the SSH server tries to authenticate by generating a challenge using the public key. Only the private key can answer this challenge. As the keys are related, log-in is successful.

Prerequisites for Auto-Save and Copy Router Configuration Using Public Key Authentication

Ensure you have enabled public key-based authentication of SSH clients, using the following steps:

- Generate RSA key pair on the router configured as the SSH client. Use the **crypto key generate authentication-ssh rsa** command to generate the RSA key pair.
- Use the **show crypto key mypubkey authentication-ssh rsa** command to view the details of the RSA key. The key value starts with *ssh-rsa* in this output.
- Copy the RSA public key from the SSH client to the SSH server.

For more detailed information on how to enable SSH connection using public-key based authentication, see *Public Key Based Authentication of SSH Clients* in System Security Configuration Guide for Cisco 8000 Series Routers.

Configuration Example for Auto-Save Using Public Key Authentication

When you are using public key authentication, you don't need to mention **password**.

```
Router(config-cfg-autosave)#configuration commit auto-save filename
sftp://user1@server1://test-folder/test_123
Router(config-cfg-autosave)#timestamp
Router(config-cfg-autosave)#maximum 10
Router(config-cfg-autosave)#wait-time days 0 hours 0 minutes 0 seconds 5
Router(config-cfg-autosave)#commit
```

Running Configuration

```
Router#show running-config configuration commit auto-save
configuration commit auto-save
  filename sftp://user1@server1://test-folder/test_123
  timestamp
  maximum 10
  wait-time days 0 hours 0 minutes 0 seconds 5
!
```

Increasing Commit Limit

Table 12: Feature History Table

Feature Name	Release Information	Feature Description
Increasing Commit Limit	Release 25.1.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8011-4G24Y4H-I • 8712-MOD-M

Feature Name	Release Information	Feature Description
Increasing Commit Limit	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Increasing Commit Limit	Release 24.2.11	<p>The maximum number of commits is increased in the router that allows you to configure complex topology changes without interruptions caused by the default blocking of commit changes during rebase or ASCII backup operations. You can prevent the commit operation from getting blocked by using the cfs check command, which increases the commit (pacount) count from 20 to 40, and the commit file diff size (configuration data) from 2 MB to 4 MB, and by using the clear configuration ascii inconsistency command, which performs an ASCII backup after 55 minutes.</p> <p>The feature modifies the following commands:</p> <ul style="list-style-type: none"> • cfs check • clear configuration ascii inconsistency

The Cisco IOS XR Routers use a two-stage configuration model. The first stage is target configuration, where you build the configurations using necessary commands in the command line interface. The second stage is the commit, where the configuration made in the target stage is added to the router configuration using the **commit** command. After each commit, the router generates a file for the newly configured changes and adds it to its running configuration, making it an integral part of the running configuration.



Note This target configuration does not impact the router's running configuration.

The Cisco IOS XR routers perform rebase and ASCII backup operations to maintain the real time configuration in the backup copy. The rebase and ASCII backup operations block you from committing configurations to the router.



Note Starting with Release 24.3.1, the rebase operation no longer blocks the commit operation.

This allows you to configure complex topology changes without being interrupted by the default blocking of commit changes during the rebase operation. For more information, see the section [Concurrent Configuration Rebase during Commit, on page 105](#).

In rebase, the router automatically saves your changes to the backup binary configuration file after 20 commits, or 2 MB of configuration data. The router blocks the commit while saving the configuration to the backup file. The router takes a few seconds to complete the rebase operation, during which, if you terminate the CLI session, the router loses the target configurations in the blocked commit.

In ASCII backup, the router automatically saves a copy of its running configuration in the ASCII format. This backup process takes place if there has been a commit to the router configuration and when the ASCII backup timer completes a 55-minute window after the previous backup event. However, if there was no commit when the ASCII backup timer completes 55 minutes, the counter is reset without any backup. During the ASCII backup, the router blocks the configuration commits.

Starting with Release 24.2.11, we have made the following enhancements:

- You can use the **cfs check** command to increase the rebase limits in the router from 20 to 40 commits and the configuration data from 2 MB to 4 MB. When configuring the router, you can check the current commit count and configuration data size using the **show cfmgr commitdb** command. If the commit count is 20 or higher, or the configuration data size is 2 MB or above, the router will initiate a rebase within 10 seconds. By using the **cfs check** command to increase the commit count to 40 and the configuration data to 4 MB, you can commit without delay.
- You can use the **clear configuration ascii inconsistency** command to perform an ASCII backup and reset the ASCII backup timer to zero. Once the backup is complete, the router will automatically initiate the next periodic ASCII backup operation only after 55 minutes from the time the **clear configuration ascii inconsistency** command is executed.

Guidelines and Restrictions for Increasing the Commit Limit

- The **clear configuration ascii inconsistency** command initiates an ASCII backup and resets the ASCII backup timer count to zero. Following this, the router will automatically initiate the next periodic ASCII backup operation only after 55 minutes from the time **clear configuration ascii inconsistency** command is executed. For example, if you execute a commit operation after executing a **clear configuration ascii inconsistency** command, the router will perform an ASCII backup operation 55 minutes after the **clear configuration ascii inconsistency** command was executed, and merge the new commit into ASCII backup. Hence, before the next 55 minutes, you must execute the **clear configuration ascii inconsistency** command again to reset the ASCII backup timer to zero.

- When the router enters standby mode or reloads, the ASCII timer does not reset to zero, and the router performs an ASCII backup operation 55 minutes after the first commit operation before the standby mode or reload.
- Cisco does not recommend executing **clear configuration inconsistency** and **clear configuration ascii inconsistency** commands regularly after each commit, as it causes hard disk wear and tear. You should execute these commands only before a commit or sequence of commits that must be done within a specific timeframe and without being delayed by rebase and ASCII backup operations. As these commands perform disk input and output operations in the background, frequent execution of these commands causes frequent access to the hard disk, which increases the wear and tear on the hard disk.

Increasing the Rebase Limits

You can increase the rebase limits as follows:

1. Use the **cfs check** command to increase the commit count to 40 and the configuration data to 4 MB.

```
Router# cfs check
Creating any missing directories in Configuration File system...OK
Initializing Configuration Version Manager...OK
Syncing commit database with running configuration...OK
```

2. Verify if the **cfs check** command is executed using the **show configuration history** command.

```
Router# show configuration history last 5
Sno.  Event      Info                               Time Stamp
~~~~  ~~~~~      ~~~~                               ~~~~~
1     cfs check  completed                          Wed Jan 10 11:42:21 2024
2     commit    id 1000000001                       Wed Jan 10 11:39:26 2024
3     startup   configuration applied                 Wed Jan 10 11:39:02 2024
```

Perform ASCII Backup and Rest ASCII Backup Timer

You can perform ASCII backup and rest ASCII backup timer as follows:

1. Use the **clear configuration ascii inconsistency** command to perform ASCII backup at that instance and reset the ASCII backup timer count to zero.

```
Router# clear configuration ascii inconsistency
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!! It is recommended to run this command only when all nodes in router      !!!!
!!!! are in IOS-XR RUN state. To determine node state, run following command: !!!!
!!!! 'show platform'.                                                         !!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Warning: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Proceed with the command?[confirm] y
Ascii configuration backup is in progress...
Configuration ascii backup complete
```

2. Verify if the **clear configuration ascii inconsistency** command is executed using the **show configuration history** command.

```
Router# show configuration history last 5
Sno.  Event      Info                               Time Stam
~~~~  ~~~~~      ~~~~                               ~~~~~
1     backup     Periodic ASCII backup              Wed Jan 10 11:48:20 2024
2     cfs check  completed                          Wed Jan 10 11:42:21 2024
3     commit    id 1000000001                       Wed Jan 10 11:39:26 2024
4     startup   configuration applied                 Wed Jan 10 11:39:02 2024
```

Concurrent Configuration Rebase during Commit

Table 13: Feature History Table

Feature Name	Release Information	Feature Description
Concurrent Configuration Rebase during Commit	Release 25.1.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8011-4G24Y4H-I • 8712-MOD-M
Concurrent Configuration Rebase during Commit	Release 24.3.1	<p>Introduced in this release on: Fixed Systems (8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100])</p> <p>The router performs the commit and rebase operations simultaneously, ensuring that the commit operation remains unblocked during the rebase operation.</p> <p>This removes the need to use the cfs check command to increase the commit count and the commit file diff size.</p>

Cisco IOS XR routers use a two-stage configuration model. In the first stage, configurations are built using necessary commands in the command line interface, and in the second stage, the configurations are committed to the router.

During rebase and ASCII backup operations, the router blocks configuration commits. However, the "Concurrent Configuration Rebase during Commit" feature allows the router to perform commit and rebase operations simultaneously, ensuring that the commit operation remains unblocked during the rebase operation.

The Cisco IOS XR routers perform rebase and ASCII backup operations to maintain the real time configuration in the backup copy.

Before Release 24.3.1,

- The rebase and ASCII backup operations block you from committing configurations to the router.
- You can increase the maximum number of commits and reset the ASCII backup timer to allow the router to configure complex topology changes without interruptions caused by the default blocking of commit changes during rebase or ASCII backup operations. For more information, see the section [Increasing Commit Limit, on page 101](#).

From Release 24.3.1,

- The router performs the commit and rebase operations simultaneously, ensuring that the commit operation remains unblocked during the rebase operation. This removes the need to use the **cfs check** command to increase the commit count and the commit file diff size.
- However, the ASCII backup operations still block the commit operation. You can reset the ASCII backup timer using the `clear configuration ascii inconsistency` command. This allows the router to perform an ASCII backup after 55 minutes and perform commit operations without being blocked by ASCII backup operations. For more information on ASCII backup, see the section [Increasing Commit Limit, on page 101](#).

View VRF-specific Configuration

Table 14: Feature History Table

Feature Name	Release Information	Feature Description
View VRF-specific Configuration	Release 25.1.1	<p>Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8011-4G24Y4H-I • 8712-MOD-M
View VRF-specific Configuration	Release 24.4.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*)</p> <p>*This feature is supported on:</p> <ul style="list-style-type: none"> • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
View VRF-specific Configuration	Release 24.2.11	<p>You can now filter the configurations associated with a specific VRF using the show running-configuration filter vrf command. Earlier, the show running configuration command displayed configuration under a specific keyword only and that may not publish all configurations related to the object.</p> <p>CLI:</p> <p>show running-configuration filter vrf</p>

Starting Cisco IOS XR Software Release 24.2.11, you can use the **show running-configuration filter vrf** command publishes all running configuration data associated with a VRF name, including configurations under various CLI link-points or keywords. Also, it reveals logical dependencies, showing parent interface configurations for subinterfaces and bundle configurations for member interfaces that are part of a VRF.

Restrictions for Filtering Running Configuration for VRF

- The **show running-configuration filter vrf** doesn't support filtering for default VRFs that aren't explicitly labeled in the configuration.
- Logical dependencies are only automatically displayed for main interfaces with their subinterfaces and bundle interfaces with their members; other dependencies require a specific request.
- Only a subset of the data from the **show running configuration** command is displayed by the new CLI.
- The accuracy of the data provided by the new CLI isn't independently verified.

View Filter Running Configuration for VRF

Procedure

Use the **show running-configuration filter vrf** command to filter the running configuration for a specific vrf and list all the configurations related to that object.

Example:

```
Router#show running-configuration filter vrf vrf80

!! Building configuration...
!! IOS XR Configuration 24.2.1.32M
!! Last configuration change at Mon Jan 15 05:09:20 2024 by cisco
!
vrf vrf80
  address-family ipv4 unicast
    import route-target
```

```
    1:80
    !
    export route-target
    1:80
    !
    !
    address-family ipv6 unicast
    import route-target
    1:80
    !
    export route-target
    1:80
    !
    !
    !
    neighbor 192.0.2.1
    remote-as 200
    ebgp-multihop 4
    update-source Loopback90
    address-family ipv4 unicast
    route-policy PASS in
    route-policy PASS out
    !
    !
    !
    !
end
```

Eco Mode Power Saving for fabric and NPU

Eco mode is a power management feature that

- enables users to reduce the power consumption of supported hardware components
- allows simplified global or selective activation of power-saving settings, and
- provides granularity to manage power modes at both the system-wide and feature-specific level.

Table 15: Feature History Table

Feature Name	Release Information	Feature Description
Eco Mode Power Saving for fabric and NPU	Release 25.3.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100])</p> <p>You can reduce overall hardware power consumption and streamline energy management by enabling eco mode for supported components like fabric and NPU. With this feature, you can configure power-saving settings globally or tailor them for specific subcategories and features. Eco mode allows hardware components to operate in their lowest supported power states, minimizing manual steps and helping organizations optimize power usage across the system.</p>

Streamline hardware power settings using eco mode

Eco mode streamlines the process of setting hardware components such as the fabric and network processing unit (NPU) to their lowest supported power states for energy efficiency.

When eco mode is enabled, the system automatically applies the lowest supported power modes to all applicable hardware components or to selected subcategories and features. Users can enable eco mode globally or tailor it for specific subcategories (such as fabric or NPU) and their features. Eco mode reduces manual steps and helps organizations optimize power usage across the platform.

Guidelines for using eco mode and power profiles

- Set explicit power profiles when you require specific performance levels, because these settings override eco mode.
- Use global eco mode for maximum simplicity and power savings.
- Disable eco mode for individual features only if operational requirements demand it.
- Reload the card after making changes to eco mode settings to ensure the new settings take effect.

Configure eco mode power saving

Reduce system power consumption by activating eco mode globally or for specific components and features.

Previously, reducing power usage required manual setting of power profiles for all fabric card and line cards. The PSM module automates and standardizes eco mode configuration, reducing operational effort and improving consistency. Eco mode can be set globally or with subcategory and feature-level granularity; explicit NPU power-profile configuration always takes precedence.

Procedure

Step 1 Enable eco mode globally.

Example:

```
Router# hw-module power-saving-mode eco enable
```

This action sets all subcategories and features to their lowest supported power state, automatically updating fabric card and line cards based on the PSM configuration.

Step 2 Enable eco mode for a specific subcategory and feature.

Selecting subcategory NPU enables eco mode on line cards, and selecting fabric enables eco mode on fabric cards.

Example:

```
Router# hw-module power-saving-mode eco sub-category npu feature power-profile enable
```

```
Router# hw-module power-saving-mode eco sub-category fabric feature all enable
```

Replace power-profile or all with the target feature for your hardware. Granular enablement lets you selectively apply eco mode on supported features.

Step 3 Disable eco mode for a specific subcategory and feature (requires global eco mode to be enabled first).

Example:

```
Router# hw-module power-saving-mode eco sub-category fan feature feature_name disable
```

Disabling eco mode at the feature level allows exceptions if global eco mode applies to all by default. Replace *feature_name* with your specific feature.

Step 4 Set the NPU power profile explicitly.

Choose **high**, **medium**, or **low** for NPUs on respective card types. When configured, this overrides eco mode settings for power profile.

Example:

```
Router(config)# hw-module npu-power-profile card-type LC high
```

```
Router(config)# hw-module npu-power-profile card-type FC medium
```

Step 5 Reload the hardware card to apply the eco mode changes.

Example:

```
Router# hw-module reload card <card_slot_number>
```

After configuration, reload the relevant fabric card, NPU, or line card. Replace <card_slot_number> with your actual card slot identifier.

Step 6 Verify components are operating in eco mode and review power profile status after reload.

Example:

```
Router# show hw-module power-saving-mode status
```

```
Router# show controllers npu proc-cmd "pi get_power_profile" location 0/RP0/CPU0
Router# show controllers npu slice info slice all location 0/RP0/CPU0 | i NPU Core
Router# show env power
Router# show controller power-saving-mode boot-config
```

Selected components and features operate in their lowest supported power-saving mode. Explicit NPU power profiles (if set) take precedence over eco mode. Power savings and operational status can be monitored and verified using system show commands.

Monitor mount points

The mount-points is a container in the OpenConfig YANG model that

- monitors the list of mount points or file systems on the router
- provides details for each file system, including its name, storage component, type, total size, available, and utilized space, and
- supports the streaming of operational data through Model-Driven Telemetry (MDT) and Event-Driven Telemetry (EDT).

Table 16: Feature History Table

Feature Name	Release Information	Feature Description
Monitor mount points	Release 26.1.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: Q100, Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100])</p> <p>You can prevent service disruptions by having a clear visibility into your router's storage resource allocation and usage. Adequate disk space is crucial for preventing application crashes, ensuring successful upgrades, and maintaining optimal router performance.</p> <p>This feature provides detailed information about the file system's name (mount point), storage component (file system type), total size, available space, and utilized space, allowing you to monitor and manage disk space effectively.</p> <p>This feature introduces these changes:</p> <p>CLI:</p> <ul style="list-style-type: none"> The detail keyword is added to the show filesystem command. <p>YANG Data Model:</p> <ul style="list-style-type: none"> openconfig-system:system/mount-points <p>Both Model-Driven Telemetry (MDT) and Event-Driven Telemetry (EDT) are supported. (See GitHub, YANG Data Models Navigator)</p>

Restrictions for mount-point state monitoring

You must follow these restrictions when performing monitoring or operational reporting from mount points:

- Mount-point state monitoring is supported only on these file systems: `ext2`, `ext4`, `tmpfs`, `devtmpfs`, `fuse.tftp_fs`, and `fuse.ftp_fs`.
- If you create mount points dynamically, you must create them within the XR namespace to:
 - enable streaming through the OpenConfig system model, and
 - ensure display in the `show filesystem detail` CLI output.

For example, create mount points using `PID 1` namespace inheritance by running `nsenter -t 1`. This command ensures the mount point is created in the XR namespace.



CHAPTER 10

Process Memory Management

This module describes memory management processes.

In Cisco IOS XR routers, process memory management refers to the efficient allocation and isolation of memory for individual processes within the modular operating system. Each process runs in its own protected memory space to ensure stability and prevent interference.

IOS XR routers has dynamic memory allocation, leveraging segments like code, data, stack, heap, and shared memory for process execution.

- [Core dump folder limit, on page 113](#)

Core dump folder limit

A core dump folder is a designated directory in the Cisco IOS XR file system that:

- stores diagnostic files, known as **core dumps**, which capture the state of the system's memory during a crash or failure,
- is typically located at `/misc/disk1/coredumps`, and
- provides critical information for troubleshooting and debugging system issues.

To manage storage effectively and ensure system reliability, Cisco IOS XR introduces mechanisms such as:

- configurable storage limits to control the amount of space used by core dumps,
- periodic cleanup to remove older files and free up disk space, and
- syslog alerts to notify administrators of storage issues, optimizing disk usage and preventing data loss during repeated crash scenarios.

Table 17: Feature History Table

Feature Name	Release	Description
Core dump folder limit	Release 25.2.1	<p>Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100, K100])</p> <p>You can now set configurable disk storage limits between 20% and 50% of total space to prevent excessive core dumps during repeated crashes. This is complemented by a new core dump folder limit, which restricts the number of retained coreinfo files to a maximum of 50. Together, these features ensure better disk management by controlling storage consumption and preserving space for critical system operations. Additionally, periodic automatic cleanup every 15 minutes helps maintain optimal disk usage, enhancing overall system reliability and performance.</p> <p>CLI:</p> <ul style="list-style-type: none"> The disk-usage-limit keyword is added to the exception command. <p>YANG Data Model:</p> <ul style="list-style-type: none"> <code>Cisco-IOS-XR-sysadmin-dumper.yang</code> <p>(see GitHub, YANG Data Models Navigator)</p>

Before Release 25.2.1, the core dump folder lacked a storage limit for core dump files,

- which could result in the allocated hard disk space being filled during continuous crash scenarios.
- This situation could prevent the Cisco IOS XR software from saving the latest core files, leading to data loss.

From Release 25.2.1:

- You can configure the following to effectively control the core dump folder storage:
 - **Throttling mechanism bypass:** Use the **exception filepath** to redirect core files to another directory, utilizing new storage space.
 - **Disk usage limit:** Use the **exception disk-usage-limit <usage limit percent>** command to set the storage limit between 20% and 50% of the total hard disk space.

Periodic cleanup

Before Release 25.2.1, the Cisco IOS XR software generate a syslog message when the disk was full without performing any cleanup.

From Release 25.2.1, The Cisco IOS XR software automatically performs a periodic check every 15 minutes and deletes older coreinfo files if the count in core dump folder exceeds 50.

This cleanup specifically targets coreinfo files such as files ending with core.txt.

During a cleanup scenario, the Cisco IOS XR software displays the following syslog message:

```
Router:Jan 20 20:15:02.014 UTC: dumper[69243]: %OS-SYSLOG-6-LOG_INFO : Coreinfo count is
currently 52, deleting
/misc/disk1/coredumps/sleep_11899.by.3.20250120-191744.xr-vm_node0_RP0_CPU0.34684.core.txt

RP/0/RP0/CPU0:Jan 20 20:15:02.056 UTC: dumper[69255]: %OS-SYSLOG-6-LOG_INFO : Coreinfo count
is currently 51, deleting
/misc/disk1/coredumps/sleep_13995.by.3.20250120-193249.xr-vm_node0_RP0_CPU0.34684.core.txt
```

Restrictions of core dump folder limit

- **Core file saving abort scenarios:**

The system displays syslog messages instead of saving host core dumps when it encounters the following storage constraints:

- **Insufficient Disk Space less than 10 percent:**

If the available disk space is less than threshold limit of 10 percent, the Cisco IOS XR software aborts the core file copy and deletes the file from the temporary directory.

In this syslog example, the Cisco IOS XR software aborts the core file copy when the space in the hard disk is less than 10 percent.

```
Router:Jan 31 10:18:42.000 UTC: dumper[1759]: %INFRA-CALVADOS_DUMPER-2-ABORT_COPY
: Copy of sleep_16135.by.3.20250131-101800.ios.34684.core.lz4 to
0/RP0:/misc/disk1/coredumps aborted for reason: Not enough available space in
/misc/disk1 (less than 10%). Deleting core from /misc/scratch/core
```

- **Exceeding Disk Usage Limit:**

If the core dump folder storage size exceeds the configured disk usage threshold limit and the latest core file in the hard disk is within 10 minutes than the current crash time, the Cisco IOS XR software

- aborts saving of the new core file and
- deletes incoming core files.

In this syslog example, the Cisco IOS XR software aborts the core file copy when the disk usage exceeds 20 percent of the hard disk's threshold value and the latest core file in the hard disk is within 10 minutes than the current crash time.

```
Router :Mar 25 05:53:08.000 UTC: dumper[2715]: %INFRA-CALVADOS_DUMPER-2-ABORT_COPY
: Copy of sleep_21216.by.3.20380325-055224.ios.34684.core.lz4 to
0/RSP0:/misc/disk1/coredumps aborted for reason: Dumper disk usage exceeds 20%
threshold of /misc/disk1. Deleting core from /misc/scratch/core
```

How core dump folder limit works

Summary

The process of managing core dump storage involves:

- Monitoring disk usage against a configured threshold.
- Deleting older core files when the disk usage exceeds the limit and conditions allow.
- Aborting saving of new core files if the disk usage is critically high or if conditions prevent older files from being deleted.

The key components that are involved in core dump folder limit are:

- **Cisco IOS XR Software:** Monitors disk usage and performs actions.
- **Core dump folder:** Stores the core files created during system crashes.

Workflow

After enabling the disk usage limit with the **exception disk-usage-limit <usage limit percent>** command, IOS XR software performs these actions to control the core dump folder storage under various disk storage scenarios:

1. Cisco IOS XR software continuously monitors the disk usage of the core dump folder against the configured limit.
2. If the core dump folder storage size exceeds the configured disk usage limit value, Cisco IOS XR software performs the following actions:

When...	And...	Then Cisco IOS XR software...	And displays these syslog message...
the core dump folder storage size exceeds the configured disk usage limit value and	if the latest core file in the hard disk is more than 10 minutes older than the current crash time	deletes the older core files starting from oldest to newest until the core dump folder storage value is below the default or configured disk usage limit value.	<pre> Router:Jan 21 10:19:36.119 UTC: dumper[67458]: %OS-SYSLOG-3-LOG_ERR : sleep_14157 signature 4f24a3679e430a4a68d6096785c26548 RP/0/RSP0/CPU0:Jan 21 10:19:36.251 UTC: dumper[399]: %OS-COREFILTER-6-DELETE_STALE_CORE : Deleting /misc/disk1/coredumps/ sleep_10432.by.3.20250121-100007 xr-mrnode0_RSP0_CPU0_34684.core.lza on active RP for reason: Freeing up old core in /misc/disk1/coredumps, attempting to copy new core file </pre>

When...	And...	Then Cisco IOS XR software...	And displays these syslog message...
the core dump folder storage size exceeds the configured disk usage limit value and	if the latest core file in the hard disk is within 10 minutes than the current crash time	will abort saving the core files and delete any incoming core files.	Router:/misc/disk1/coredumps] \$RP/0/RSP0/CPU0: Jan 24 10:23:53.921 UTC: dumper[175]: %OS -COREHELPER-2-ABORT_COPY : Copy of sleep_19617.by.3. 20250124-102353.xr-vm_node0_RSP0_CPU0.34684.core.lz4 to 0/RSP0/CPU0:/misc/disk1/coredumps aborted for reason: Dumper disk usage exceeds 20% threshold of /misc/disk1. Potential /misc/disk1/coredumps usage with new core: 100.0%, seconds since last core: 36.0s. Deleting core from /misc/scratch/core.
the hard disk is filled more than 90 percent of the total hard disk space	-	will abort saving the core files and delete any incoming core files as it can't delete any old core files.	Router:Mar 25 04:25:06.252 UTC: dumper[329]: %OS -COREHELPER-2-ABORT_COPY : Copy of sleep_30731.by.3.20380325-042505.xr-vm_node0_RSP0_CPU0.34684.core.lz4 to 0/RSP0/CPU0:/misc/disk1/coredumps aborted for reason: Not enough available space in /misc/disk1 (less than 10%). Current /misc/disk1 bytes in use: 23590813696B, size of core file: 1471276B. Deleting core from /misc/scratch/core.

Configure core dump folder limit

You must perform these steps to control the core dump folder storage limit:

Procedure

Step 1 Set the core dump folder storage limit to prevent excessive core dumps on the hard disk.

Example:

```
Router# configuration
Router(config)# exception disk-usage-limit 29
Router(config)# commit
```

You can configure the core dump folder storage limit to range between 20 and 50 percent of the total hard disk space.

Step 2 Run **show exception** command to view the set core dump folder storage limit.

Example:

This example displays the core dump folder storage limit set to 29 percent of the total hard disk space.

```
Router# show exception
Exception path for choice 1 is not configured or removed
Exception path for choice 2 is not configured or removed
Exception path for choice 3 is not configured or removed
Default fallback/copy path = /misc/disk1/coredumps
Core dump usage on disk limited to 29%
```

Example:

This example displays the core dump folder storage limit reset to 20 percent of the total hard disk space after disabling the **exception disk-usage-limit** command.

```
Router# show exception
Exception path for choice 1 is not configured or removed
Exception path for choice 2 is not configured or removed
Exception path for choice 3 is not configured or removed
Default fallback/copy path = /misc/disk1/coredumps
Core dump usage on disk limited to 20%
```
