System Management Configuration Guide for Cisco 8000 Series Routers, Cisco IOS XR Releases

Cisco 8000 Series Routers
Release: Cumulative | Updated September 11, 2026

1 What's changed in System Management Configuration Guide

Topics:

- Getting Started

Summarizes the release updates and feature changes for the System Management Configuration Guide. It offers a single, continuously updated version of the documentation to simplify navigation across multiple release-specific versions.

Getting Started

Outlines the release history and update summary for the System Management Configuration Guide. It enables users to access the latest IOS XR features and release information through a centralized, continuously updated resource.

This cumulative guide provides a single, continuously updated version that includes all the latest IOS XR features and release updates. It simplifies your experience by letting you bookmark one link and access the complete guide, instead of navigating through multiple release-specific versions.

Specific changes or updates tied to individual releases are clearly called out within the relevant sections. For a list of features introduced in a specific release, refer to the Release Notes.

2 Cisco Discovery Protocol

Topics:

- Cisco Discovery Protocol Version 2
- How CDP operates
- Enable CDP
- Modify CDP default settings
- Monitor CDP

Describes the Cisco Discovery Protocol (CDP) and explains how to enable, configure, and monitor the protocol on Cisco devices to facilitate network topology discovery and device management.

(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.

Cisco Discovery Protocol Version 2

Describes Cisco Discovery Protocol version 2 (CDPv2), which provides enhanced device tracking, rapid error reporting for network mismatches, and detailed output for VLAN and duplex state management.

Cisco Discovery Protocol version 2 (CDPv2) is a network discovery protocol that:

- provides enhanced device tracking and identification features,
- enables rapid error reporting on network mismatches and configuration issues, and
- supports detailed output on VLAN and duplex state management between connected devices.

You can use CDPv2 to quickly report errors, such as when native VLAN IDs (IEEE 802.1Q) do not match or when port duplex states are inconsistent.

CDPv2 show commands display detailed output about the VLAN Trunking Protocol (VTP) management domain, duplex modes of neighboring devices, CDP-related counters, and VLAN IDs of connected ports.

Definitions of CDP type-length-value fields

Provides definitions for Type-Length-Value (TLV) fields, which are information blocks embedded in CDP advertisements that identify devices, addresses, ports, capabilities, software versions, and hardware platforms.

Type-length-value fields (TLVs) are information blocks embedded in CDP advertisements. This table provides TLV definitions for CDP advertisements.

Table 1: 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.

Features of CDP

Lists the primary capabilities of CDP, including obtaining neighbor device addresses, displaying interface information, and operating at the data link layer across various media types.

CDP provides the following capabilities:

- Obtains protocol addresses and platform identity of neighboring devices.
- Displays information about router interfaces and connected devices.
- Operates media- and protocol-independently on all Cisco platforms: routers, bridges, access servers, switches.
- Uses SNMP with CDP MIB (CISCO-CDP-MIB) for network management and device querying.
- Runs on media that support Subnetwork Access Protocol (SNAP), such as LAN, Frame Relay, ATM.
- Functions solely at the data link layer, enabling systems with differing network-layer protocols to communicate.

How CDP operates

Describes the CDP process, which involves periodic exchange of advertisement messages to enable network devices to discover neighbors and maintain up-to-date network topology information.

Summary

The CDP process enables network devices to discover neighboring devices and maintain up-to-date network topology information using periodic message exchanges.

The key components involved in the process are:

- CDP-enabled device: Sends and receives CDP advertisement messages to identify network neighbors.
- Multicast address: Receives CDP advertisements broadcast by devices.
- Hold-time parameter: Specifies how long CDP information should be retained before it is discarded.

Workflow

The process involves these stages:

1. Each CDP-enabled device periodically sends advertisement messages containing its SNMP address and hold-time information to a designated multicast address.
2. Neighboring devices listen for incoming CDP advertisements to learn about nearby devices and their capabilities.
3. Each receiving device stores CDP information and maintains it based on the hold-time value provided in the advertisements.
4. Devices track network changes by observing new, updated, or missing CDP messages. This approach allows them to determine when interfaces go up or down.

Result

CDP devices maintain an updated view of their network neighborhood, enabling efficient network management, troubleshooting, and topology discovery.

Enable CDP

Teaches you how to enable CDP globally and on specific interfaces to ensure your router shares device information with directly connected Cisco equipment.

Enable CDP so that the router shares device information with other directly connected Cisco devices.

To enable CDP, first enable it globally on the router. Then enable CDP on each interface.

Before you begin

Verify that the CDP package is installed on your router.

Procedure

1. Enable CDP globally.

   Example

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

2. Enable CDP on the specific interface.

   Example

   ```
   Router#configure
   Router(config)#interface hundredGigE 0/0/0/4
   Router(config-if)#cdp
   Router(config-if)#commit
   ```

Modify CDP default settings

Use this procedure to customize CDP settings for compatibility and network discovery frequency.

Ensure CDP operates with customizable settings for compatibility and desired frequency of network discovery information.

Changing CDP defaults allows you to improve compatibility and tuning for network discovery needs.

Before you begin

CDP must be enabled on the router.

Procedure

1. Configure CDP to use only version 1.

   Example

   ```
   Router#configure
   Router(config)#cdp advertise v1
   ```

2. Set the time the receiving device holds a CDP packet before discarding.

   Example

   ```
   Router(config)#cdp holdtime 30
   ```

> **Note**
>
> Default hold time is 180 seconds.
>
> Ensure the hold time is greater than the CDP timer value.

3. Set the frequency at which CDP update packets are sent.

Example

```
Router(config)#cdp timer 20
```

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.

Monitor CDP

Use this procedure to monitor Cisco Discovery Protocol (CDP) neighbor and interface information using show commands.

Monitor the neighboring devices and the CDP-enabled interfaces on your network.

Use these commands to view information about CDP-related devices, interfaces, neighbors, and traffic.

Procedure

1. View details about specific or all neighboring devices.

Example

```
Router#show cdp entry
-------------------------
Device ID: chiba
SysName : chiba
Entry address(es):
Platform: cisco 8000,  Capabilities: Router
Interface: TenGigE0/2/0/19
Port ID (outgoing port): TenGigE0/0/0/2/0
Holdtime : 171 sec

Version :
26.2.1.19I

advertisement version: 2
Duplex: full
```

2. Display information about the interfaces on which CDP is enabled.

   Example

```
Router#show cdp interface
Tue Jan 20 17:57:15.108 EST
TenGigE0/2/0/19 is Up
  Encapsulation ether
  Sending CDP packets every 10 seconds
  Holdtime is 180 seconds
```

3. View information about all neighboring devices.

   Example

```
Router# show cdp neighbors
Tue Jan 20 17:58:36.451 EST
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
chiba           Te0/2/0/19       172     R          8000      Te0/0/0/2/0
```

4. Display CDP traffic statistics.

   Example

```
Router#show cdp traffic
CDP counters :
        Packets output: 61, Input: 58
        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: 61, Input: 58
        Unrecognize Hdr version: 0, File open failed: 0
```

3 Physical and Virtual Terminals

Topics:

- Physical terminal lines
- Virtual terminals
- Line templates
- Configure the physical and virtual terminals

Describes the implementation of physical console ports and virtual terminal lines (vtys) using line templates and vty pools to manage attribute settings for network device access in Cisco IOS XR.

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

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

Physical terminal lines

Use this concept to understand how physical terminal lines identify and enable access to device console ports based on specific hardware location within a system.

A physical terminal line is a network device interface that

- is identified by its precise location, expressed in the format of rack/slot/module,
- resides on either the active or standby route processor (RP) of the system, and
- provides direct access to the device's console port for management and troubleshooting.

Physical terminal lines are commonly used for out-of-band device management, allowing administrators to connect directly even if network services are unavailable.

A typical physical terminal line identifier might be 0/0/CPU0 for a console port located in rack 0, slot 0, and module CPU0 on the active route processor.

Virtual terminals

This section explains virtual terminals, which enable remote access and administration of the network device.

A virtual terminal is a router management interface that

- provides remote access to the device using Telnet or SSH,
- operates independently of physical connections, and
- is identified by a unique vty identifier assigned according to connection order.

vty lines: The software-maintained logical lines that facilitate multiple virtual terminal sessions.

- Physical location is not applicable for virtual terminals, as all sessions are established virtually.
- The Cisco IOS XR software assigns a vty identifier to each vty session based on establishment sequence.

Connecting to a router using **ssh admin@router-ip** uses a virtual terminal line.

Vty pools

Use this concept to understand how vty pools manage network device remote access by grouping vty lines and applying line templates for consistent connection policies.

A vty pool is a connection management construct that

- groups multiple virtual terminal lines (vtys) into logical pools,
- applies a common line template configuration to all member vtys, and
- allows flexible division of remote access connections for role- or service-based segmentation.

- Default vty pool: Consists of five vtys (vtys 0 through 4) that each reference the default line template.
- Default fault manager pool: Consists of six vtys (vtys 100 through 105) that each reference the default line template.
- User-defined vty pool: In addition to the default vty pool and default fault manager pool, you can configure a user-defined vty pool that can reference the default template or a user-defined template.

Guidelines for vty Pools

This section outlines recommended configuration practices for vty pools to ensure correct and efficient management of virtual terminal lines.

- The vty range for the default pool must start at vty 0 and include at least five vtys.
- The vty range from 0 through 99 can reference the default vty pool.
- Resize the default vty pool by increasing the range of vtys as needed.
- The vty range from 5 through 99 can reference a user-defined vty pool.
- If the default vty pool is resized, start the next available user-defined pool at the first free vty (e.g., after resizing vty 0-9 for the default, begin the next at 10).
- The vty range from 100 is reserved for the fault manager vty pool, which must start at vty 100 and include at least six vtys.
- A vty can belong to only one pool; configuration fails if a vty is assigned elsewhere.
- Attempting to remove an active vty from its pool during configuration causes the operation to fail.

Line templates

This section explains line templates, which simplify the configuration and management of multiple terminal lines in Cisco IOS XR software.

A line template is a configuration mechanism that

- defines a standard set of attribute settings for terminal lines,
- enables consistent application of those settings to multiple physical and virtual terminal lines, and
- simplifies configuration management by reducing repetitive commands.

These line template types are available in Cisco IOS XR software:

- Default line template: Applies to all physical and virtual terminal lines.
- Console line template: Applies to the console line.
- User-defined line template: Applies to designated virtual terminal line ranges.

Line Templates Types

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 Template

User-defined line templates that can be applied to a range of virtual terminal lines.

Line template guidelines

This section provides guidelines for configuring line templates on a router, including default values, configuration commands, and attribute inheritance.

Line templates on a router define the configuration for physical terminal lines such as the console port and virtual terminal lines such as vty lines used for remote access. The following information outlines how to configure these templates and the default values applied.

- To configure the console line template, enter line template configuration mode using the **line console** command from Global Configuration mode.
- To configure a virtual line template, create a user-defined template with the **line template-name** command. Set terminal attributes within the template, and apply the template to virtual terminal lines using the **vty pool** command.
- The default session-limit for line template is applicable to Telnet sessions only. It is not applicable for SSH sessions.

Table 2: Default template values

Attribute	Default Value	Notes
EXEC timeout	10 minutes	Applies to default line template
Width	80 characters	Applies to default line template
Length	24 lines	Applies to default line template
session-limit	Default applies	Only applicable to Telnet sessions; not SSH

Attribute Inheritance

Attributes that are not defined in the console template or any virtual template inherit values from the default template.

Configure the physical and virtual terminals

Use this task to modify the terminal attributes for the console and default line templates.

Change attributes and access methods for both physical (console) and virtual (default) terminal lines.

Physical and virtual terminals control device access for administrative users. Modifying their attributes enables you to set security and connectivity options.

Procedure

1. Modify the console line template on page 13.
2. Override the terminal settings for the default line template on page 15.
3. Create or modify vty pools on page 15.
4. Monitor terminal sessions on page 16.

Modify the console line template

Learn how to update the terminal attribute settings for the console line template to meet your access and security requirements.

Change console line behaviors such as timeouts and allowed protocols to enhance manageability and security.

Modify the console line template when you need to adjust session-related parameters for users who access the device via the console port. This may include setting session limits, timeouts, or specifying permitted transport protocols.

Procedure

1. Access the console line configuration

   Example

   ```
   Router#configure
   Router(config)#line console
   ```

2. Set the exec timeout to unlimited.

   Example

   ```
   Router(config-line)#exec-timeout 0 0
   ```

3. Define the escape character as hexadecimal 0x5a.

   Example

   ```
   Router(config-line)#escape-character 0x5a
   ```

4. Set the session limit to 10

   Example

   ```
   Router(config-line)#session-limit 10
   ```

5. Set the disconnect character to hexadecimal 0x59.

   Example

   ```
   Router(config-line)#disconnect-character 0x59
   ```

6. Set the session timeout value to 100.

   Example

   ```
   Router(config-line)#session-timeout 100
   ```

7. Allow only Telnet for input and output connections.

   Example

   ```
   Router(config-line)#transport input telnet
   Router(config-line)#transport output telnet
   Router(config-line)#commit
   ```

8. Verify the updated console line template settings.

Example

```
Router#show line console location 0/0/CPU0
```

The console line template is updated with the specified settings, applying the new attributes for future console sessions.

Override the terminal settings for the default line template

Learn how to configure the exec timeout, screen width, and screen length for the default line template on a Cisco router.

Modify the default terminal session parameters for all lines using the default template.

Perform this configuration to ensure your router sessions remain active and display the preferred amount of information.

Procedure

1. Access the default line template.

   Example

   ```
   Router#configure
   Router(config)#line default
   ```

2. Set the exec timeout to unlimited.

   Example

   ```
   Router(config-line)#exec-timeout 0 0
   ```

3. Set the terminal width.

   Example

   ```
   Router(config-line)#width 512
   ```

4. Set the terminal length.

   Example

   ```
   Router(config-line)#length 512
   Router(config-line)#commit
   ```

The default line template now has custom terminal settings applied for all new sessions.

Create or modify vty pools

Use this task to configure or update vty pools on a Cisco ASR 9000 Series Router by specifying the number of allowable Telnet servers and attaching a line template.

Configure or change vty pools to control telnet server access on the device.

vty pools determine how many concurrent telnet sessions your router can accept and allow you to set line templates for session characteristics.

Before you begin

Enable the telnet server by entering global configuration mode and using the appropriate command.

Procedure

1. Enable and specify the maximum number of telnet servers (up to 100).

   Example

```
Router#configure
Router(config)#telnet ipv4 server max-servers 10
                        Router#(config)
                        line template 1
                        Router#(config-line)
                        exit
                        Router#(config)
                        vty-pool default 0 5 line-template default
                        Router#(config)
                        commit
```

2. Optional: Create a line template to define line settings:

   Example

```
Router(config)#line template 1
Router(config-line)#exit
```

3. Create or modify the vty pool and attach the line template.

   Example

```
Router(config)#vty-pool default 0 5 line-template default
Router(config)#commit
```

The router applies your configured vty pool settings, enforcing telnet server limits and session characteristics.

Monitor terminal sessions

Use this procedure to monitor terminal activity and view terminal session parameters on your router.

Monitor connected users and terminals to troubleshoot access issues or review session parameters on the router.

Monitoring terminal sessions helps network administrators verify which users are currently connected, review session attributes, and ensure proper operation of terminal lines.

Procedure

1. Display the terminal parameters of terminal lines.

Example

```
Router#show line
```

2. Display the terminal attribute settings for the current terminal line

Example

```
Router#show terminal
```

3. Displays information about the active lines on the router.

Example

```
Router#show users
```

Shows all users currently connected to the router and their session source.

4 Call Home

Topics:

- About Call Home
- Benefits of Using Call Home
- Prerequisites for Call Home
- How to Configure Call Home
- Configuring Contact Information
- Destination Profiles
- Call Home Alert Groups
- Configuring Email
- Configuring a HTTPS Proxy Server
- Sending Call-home Data through an Email
- Sending Call-home Data through HTTPS
- Configuring Call Home to use VRF
- Configuring Call Home Data Privacy
- Sending Smart License Data

Describes the Call Home feature, which provides automated email and HTTPS-based notifications for critical system events, diagnostics, and environmental faults to support efficient network management and technical assistance.

This chapter describes the configuring of the Call Home feature.

Note

Starting Cisco IOS XR software Release 25.3.1, Call Home transport mode is not supported. Configure CSLU or Smart Transport mode to ensure seamless operation of the licensing solution. For more information about configuring CSLU and Smart Transport, refer to the Cisco IOS XR Smart Licensing Using Policy.

About Call Home

Describes the functionality of Call Home, which provides email and HTTPS-based notifications for system policies, supporting various message formats for automated parsing or human review.

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

Highlights the advantages of Call Home, including multiple message formats, concurrent destinations, alert filtering, and scheduled periodic reporting for improved network maintenance.

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

Outlines the requirements for Call Home, including contact information, SMTP server details, and IP connectivity verification to ensure successful alert delivery.

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

Describes the necessary tasks for configuring Call Home, such as defining contact details, destination profiles, and alert groups, while highlighting important prerequisites for HTTPS source interface configuration in dual-stack networks.

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

Configures essential contact information for Call Home, including email addresses, contract and customer identifiers, phone numbers, and physical site details to ensure accurate identification of the message origin.

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

Procedure

1. **call-home**

   Example

   ```
   Router(config)# call-home
   Router(config-call-home)#
   ```

   Enters call home configuration mode.

2. **contact-email-addr** *email-address*

   Example

   ```
   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.

3. Optional: **contract-id** *contract-id-string*

   Example

   ```
   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 ("").

4. Optional: **customer-id** *customer-id-string*

   Example

   ```
   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 ("").

5. Optional: **phone-number** *phone-number-string*

   Example

   ```
   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.

6. Optional: **street-address** *street-address*

Example

```
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 ("").

7. Optional: **site-id** *site-id-string*

Example

```
Router(config-call-home)# site-id SJ-RouterRoom1
Router(config-call-home)# commit
```

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

8. **show call-home**

Example

```
Router# show call-home
```

Displays information about the system contacts.

Destination Profiles

Describes how destination profiles manage Call Home alerts by defining recipient lists, message formats, severity thresholds, and alert group subscriptions to ensure critical system information reaches the appropriate support channels.

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

1. **call-home**

   Example

   ```
   Router(config)# call-home
   Router(config-call-home)#
   ```

   Enters call home configuration mode.

2. **profile** *profile-name*

   Example

   ```
   Router(config-call-home)# profile my_profile
   Router(config-call-home-profile)#
   ```

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

3. **destination address email** *email-address*

   Example

   ```
   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.

4. **destination message-size-limit** *max-size*

   Example

   ```
   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.

5. **destination preferred-msg-format {short-text | long-text | xml}**

   Example

   ```
   Router(config-call-home-profile)# destination
   preferred-msg-format xml
   ```

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

6. **destination transport-method email hhtp**

   Example

   ```
   Router(config-call-home-profile)# destination
   transport-method email
   ```

   Configures the transport method for this profile.

7. **active**

   Example

   ```
   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.

8. **show call-home profile** {**all** | *profile-name*}

   Example

   ```
   Router# show call-home profile all
   ```

   Displays information about the destination profile.

Call Home Alert Groups

Describes Call Home alert groups, which categorize system events into predefined subsets to trigger targeted notifications based on severity levels and associated CLI command outputs, ensuring relevant diagnostic data reaches the appropriate support destinations.

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 or HTTP 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 inventory and configuration alert groups do not have the severity limit.

Table 3: Alert Groups and Executed Commands

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

Alert Group	Description	Executed Commands
Environmental	Events related to power, fan, and environment-sensing elements such as temperature alarms.	• **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.	• **show platform** • **show version** • **show diag** • **show inventory oid**
Syslog	Events generated by specific interesting syslog messages	• **show version** • **show logging** • **show inventory**
Configuration	User-generated request for configuration or configuration change event.	• **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 4: 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

1. **call-home**

Example

```
Router#configure
Router(config)# call-home
Router(config-call-home)#
```

Enters call home configuration mode.

2. **profile** *profile-name*

   Example

   ```
   Router(config-call-home)# profile my_profile
   Router(config-call-home-profile)#
   ```

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

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

   Example

   ```
   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.

4. **subscribe-to-alert-group syslog severity** *severity-level* **pattern** *string*

   Example

   ```
   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 ("").

5. **subscribe-to-alert-group snapshot severity** *severity-level* **pattern** *string*

   Example

   ```
   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 ("").

6. **subscribe-to-alert-group configuration severity** *severity-level* **pattern** *string*

   Example

   ```
   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 ("").

7. Use the **commit** or **end** command.

What to do next

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

Configuring Email

Configures the email server settings required for Call Home, including sender and reply-to addresses, and specifies primary and backup mail servers to ensure reliable delivery of notification messages.

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

Procedure

1. **call-home**

   Example

   ```
   Router#configure
   Router(config)# call-home
   Router(config-call-home)#
   ```

Enters call home configuration mode.

2. Optional: **sender from** *email-address*

   Example

   ```
   Router(config-call-home)# sender from
   my_email@cisco.com
   ```

   Specifies the email message “from” address.

3. Optional: **sender reply-to** *email-address*

   Example

   ```
   Router(config-call-home)# sender reply-to
   my_email@cisco.com
   ```

   Specifies the email message “reply-to” address.

4. Required: **mail-server** *address* **priority** *priority*

   Example

   ```
   Router(config-call-home)# mail-server
   198.61.170.16 priority 1
   Router(config-call-home)#commit
   ```

   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.

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

   Example

   ```
   Router#show call-home mail-server status
   ```

   Displays the status of the specified mail server.

Configuring a HTTPS Proxy Server

Configures an HTTPS proxy server for Call Home to enable secure communication and data transit through a designated proxy gateway.

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

Procedure

1. **call-home**

Example

```
Router(config)# call-home
```

Enters Call Home configuration mode.

2. **http-proxy** *proxy-server-name* **port** *port-number*

Example

```
Router(config)#http-proxy p1 port 100
```

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

Sending Call-home Data through an Email

Configures Call Home to use email as the transport method by defining destination profiles, specifying recipient email addresses, selecting message formats, and subscribing to relevant alert groups for automated event reporting.

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

Procedure

1. **call-home**

Example

```
Router(config) # call-home
```

Enters Call Home configuration mode.

2. **profile** *name*

Example

```
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.

3. **active**

Example

```
Router(config-call-home-profile) # active
```

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

4. **destination transport-method email**

Example

```
Router(config-call-home-profile) # destination transport-method email
```

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

5. **destination address email** *email-address*

Example

```
Router(config-call-home-profile) # destination address email xyz@cisco.com
```

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

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

Example

```
Router(config-call-home-profile) # destinationpreferred-msg-format xml
```

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

7. **subscribe-to-alert-group syslog severity** *severity-level* **pattern** *string*

Example

```
Router(config-call-home-profile) # subscribe-to-alert-group syslog severity
normal pattern COUNT
Router(config-call-home-profile) # commit
```

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 ("").

Sending Call-home Data through HTTPS

Configures Call Home to use HTTPS as the transport method by establishing destination profiles, setting secure URLs, and defining alert group subscriptions, while ensuring proper certificate trustpoint configuration for secure data transmission.

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

1. **call-home**

   Example

   ```
   Router(config) # call-home
   ```

   Enters Call Home configuration mode.

2. **profile** *name*

   Example

   ```
   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.

3. **active**

   Example

   ```
   Router(config-call-home-profile) # active
   ```

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

4. **destination transport-method http**

   Example

   ```
   Router(config-call-home-profile) # destination transport-method http
   ```

   Configures the message transport method for HTTPS.

5. **destination address http** *url*

   Example

   ```
   Router(config-call-home-profile) # destination address http https://example.com
   ```

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

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

   Example

   ```
   Router(config-call-home-profile) # destinationpreferred-msg-format xml
   ```

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

7. **subscribe-to-alert-group syslog severity** *severity-level* **pattern** *string*

   Example

   ```
   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 ("").

Configuring Call Home to use VRF

Configures Call Home to operate within a specific Virtual Routing and Forwarding (VRF) instance to enable network segmentation for alert data, noting that this functionality is exclusively supported for HTTPS transport methods.

Procedure

1. **call-home**

Example

```
Router(config)#call-home
```

Enters Call Home configuration mode.

2. **vrf** *vrf-name*

Example

```
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

Configures data privacy levels for Call Home messages to scrub sensitive information, such as passwords, IP addresses, or hostnames, ensuring secure alert reporting while managing potential impacts on CPU utilization.

Procedure

1. **call-home**

Example

```
Router(config) #  call-home
```

Enters the call home configuration submode.

2. **data-privacy level normal high hostname**

Example

```
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

Configures the transmission of Smart Licensing data via HTTPS by enabling reporting within a destination profile, specifying the transport method, and setting the destination URL to ensure accurate license usage reporting.

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

Procedure

1. **call-home**

   Example

   ```
   Router (config) # call-home
   ```

   Enters Call Home configuration mode.

2. **profile** *name*

   Perform either one of the below actions:

   - For sending Smart License data in TAC profile:

     ```
     Router (config-call-home) # profile CiscoTAC-1
     ```

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

     ```
     Router (config-call-home) # profile user1
     ```

3. **active**

   Example

   ```
   Router (config-call-home-profile) # active
   ```

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

4. **reporting smart-licensing-data**

   Example

   ```
   Router (config-call-home-profile) # reporting smart-licensing-data
   ```

   Enables sending Smart Licensing data.

5. **destination transport-method http**

   Example

   ```
   Router (config-call-home-profile) # destination transport-method http
   ```

   Configures the message transport method for HTTPS.

6. **destination address http** *url*

Example

```
Router (config-call-home-profile) # destination address http
https://example.com
```

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

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

Example

```
Router (config-call-home-profile) # destinationpreferred-msg-format xml
```

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

5 Periodic MIB Data Collection and Transfer

Topics:

- SNMP objects and object instances
- Bulk statistics object lists
- Bulk statistics schemas
- Bulk statistics file transfer options
- Advantages of periodic MIB data collection and transfer over the bulk file MIB
- Configure a bulk statistics object list
- Configure a bulk statistics schema
- Configure bulk statistics transfer options
- Configure bulk statistics transfer options
- Example of periodic MIB data collection and transfer

Describes the bulk statistics feature, which enables periodic collection and transfer of MIB data from routers to Network Management Systems for efficient monitoring and performance analysis.

This chapter 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.

SNMP objects and object instances

Explains the distinctions between SNMP objects and object instances, and how they are used in bulk statistics collection.

A SNMP object is a unit of SNMP management information that

- defines a specific type or class of data for network monitoring and management,
- is identified through a unique object identifier (OID), and
- can be referenced in lists for bulk statistics collection.

An object instance is a particular occurrence of a SNMP object that

- represents an individual data point or SNMP variable,
- is uniquely identified by an index or OID within a MIB table, and
- is specified in schemas to collect specific statistics.

Bulk statistics collection requires specifying both SNMP objects (types) and object instances (specific data points). MIBs, MIB tables, MIB objects, and object indices are all referenced by a series of numbers called an object identifier (OID). OIDs are used in configuring both the bulk statistics object lists (for general objects) and bulk statistics schemas (for individual object instances).

Bulk statistics object lists

Explains how bulk statistics object lists are used to group MIB objects for polling based on a shared index.

A bulk statistics object list is a user-defined grouping mechanism that

- contains MIB objects that share the same MIB index,
- allows configuration to be reused across different bulk statistics schemas, and
- enables flexible grouping regardless of MIB or MIB table membership.

Object lists are identified by a name of your choosing. All objects within a bulk statistics object list must share the same MIB index, although the objects do not need to reside in the same MIB or belong to the same MIB table. This facilitates easier configuration reuse in multiple bulk statistics schemas. For example, it's possible to group ifInOctets and a CISCO-IF-EXTENSION-MIB object in the same schema, as both are indexed by the ifIndex.

Bulk statistics schemas

Explains how bulk statistics schemas structure data selection for periodic MIB collection and transfer.

A bulk statistics schema is a data selection configuration that

- defines the object list used for data collection,
- specifies the instances of objects to retrieve (including individual or wildcard series), and
- sets the polling interval for sampling these instances.
- The default polling interval is 5 minutes.
- Each bulk statistics schema requires a unique name, which is referenced when configuring transfer options.

Bulk statistics file transfer options

Lists the available options for transferring bulk statistics files to network management stations and managing local statistics files.

Bulk statistics file transfer involves the creation and transfer of a virtual file (VFile) containing all collected data to network management stations using FTP or TFTP. The following options are available for configuring bulk statistics file transfer:

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.

- Transfer interval:
  - Specify how often the bulk statistics file is transferred.
  - Default: Once every 30 minutes.
  - The transfer interval is also used as the data collection interval for the local bulk statistics file.
- Primary and secondary destinations:
  - Configure a primary network management station to receive the statistics file.
  - Optionally configure a secondary destination, which is used if the file cannot be sent to the primary station.
- File handling after transfer:
  - After each collection period, the bulk statistics file is frozen and a new file is created to continue collecting data.
  - The frozen file is then transferred to the specified destination(s).
  - By default, the local bulk statistics file is deleted after a successful transfer to the management station.

These options allow you to manage how and where bulk statistics data is collected, stored, and transferred within your network management strategy.

Advantages of periodic MIB data collection and transfer over the bulk file MIB

Provides a comparison of the key advantages of periodic MIB data collection and transfer (bulk statistics feature) over the bulk file MIB (CISCO-BULK-FILE-MIB.my), including configuration flexibility, data integrity, and improved selection mechanisms.

Periodic MIB data collection and transfer, also known as the bulk statistics feature, offers several advantages compared to the bulk file MIB solution:

- Configuration flexibility:
  - Unlike the bulk file MIB, this feature can be configured directly through the CLI and does not require an external monitoring application.
- Target platform suitability:
  - Designed mainly for medium to high-end platforms with sufficient local storage (volatile or permanent) to store bulk statistics files.
- Improved data integrity:

    - Locally storing bulk statistics files minimizes data loss during temporary network outages.

- Advanced data selection:
    - Offers more powerful data selection features than the bulk file MIB. Allows grouping of MIB objects from different tables into data groups (object lists).

- Flexible instance selection:
    - Incorporates a more flexible instance selection mechanism–applications are not restricted to fetching an entire MIB table.

Configure a bulk statistics object list

Configure a bulk statistics object list to enable periodic SNMP MIB data collection and transfer.

Set up SNMP bulk statistics object lists as the first step for configuring periodic MIB data collection and transfer.

Periodic MIB Data Collection and Transfer enables efficient monitoring by collecting selected SNMP MIB objects. The process begins by defining one or more object lists that specify which objects to monitor.

Before you begin

- Decide which SNMP MIB objects (by name or OID) to collect.
- Confirm that all selected MIB objects share the same MIB index.

Procedure

1. Define the bulk statistics object list and enter object list configuration mode. Replace `<list-name>` with your desired list name:

   Example

   ```
   Router#configure terminal
   Router(config)# snmp-server mib bulkstat object-list ifMib
   ```

2. Add MIB objects to the object list. You can specify either an object name (if recognized) or an OID:

   Example

   To add an object by OID:

   ```
   Router(config-bulk-objects)# add 1.3.6.1.2.1.2.2.1.11
   ```

   To add an object by name:

   ```
   Router(config-bulk-objects)# add ifAdminStatus
   Router(config-bulk-objects)# add ifDescr
   ```

   Repeat the add command as necessary for each object to be monitored.

3. Confirm that all objects added to the list are indexed by the same MIB index. Objects can come from different MIBs or tables if this condition is met.

4. When specifying an object by name, ensure that the object name appears in the output of the `show snmp mib object` command.

A bulk statistics object list is configured and populated. You have defined the MIB objects—by OID or object name—to monitor for periodic SNMP MIB data collection.

Configure a bulk statistics schema

Configure a bulk statistics schema to collect periodic SNMP MIB data and control the collection interval and included object lists.

Enable periodic MIB data collection and transfer by defining a schema that determines which MIB object lists are collected, on what instances, and how frequently.

A bulk statistics schema enables SNMP-based periodic collection of user-defined MIB object lists. Configuring a schema allows you to specify the device objects to collect, the types of instances, and the polling interval.

Before you begin

Identify the object list name and instance information you want to use for data collection.

Procedure

1. Define and name the bulk statistics schema.

   Example

   ```
   Router#configure terminal
   Router#snmp-server mib bulkstat schema intE0
   ```

2. Specify the bulk statistics object list to include in this schema.

   Example

   ```
   Router(config-bulk-sc)#object-list ifMib
   ```

   Note

   Only one object list can be configured per schema. If you enter multiple `object-list` commands, the most recent entry overwrites previous ones.

3. Specify the instance information for objects to be collected. Use *only one* of these commands per schema:
   - To specify an exact instance (complete OID or interface):

     instance exact interface *sub-if* or instance exact oid *oid*
   - To specify all subindices of an OID (wild-carded instance):

     instance wild interface*sub-if* or**instance wild oid***oid-name*
   - To specify a range of instances:

     instance range start*oid***end***oid*

- To repeat for a number of MIB object instances:

  instance repetition *oid* **max** *repeat-number*

> **Note**
>
> If you configure multiple `instance` commands in a schema, the most recent command overwrites previous ones.

4. Set the polling interval in minutes for data collection:

   Example

```
Router(config-bulk-sc)# poll-interval 10
```

   The valid polling interval is from 1 to 20000 minutes; the default is 5 minutes.

The bulk statistics schema is configured. The device will now collect the specified MIB object list based on instance specification, using the defined polling interval.

Configure bulk statistics transfer options

Configure options to determine how collected bulk statistics (MIB) data are transferred from the device to a remote management station.

Set up periodic transfer of collected bulk MIB data using SNMP bulkstat, ensuring efficient file management and reliable delivery to a network management system.

Before starting, ensure you have defined both bulk statistics object lists and bulk statistics schemas. These prerequisites must be configured to enable bulk statistics transfer. The following steps guide you through configuring the transfer options for periodic MIB data collection.

Before you begin

- Bulk statistics object lists must be defined.
- Bulk statistics schemas must be defined.

Procedure

1. Define the bulk statistics transfer configuration.

   Example

```
Router# configure terminal
Router(config)# snmp-server mib bulkstat transfer bulkstat1
```

2. Optional: Set the maximum buffer size (in bytes) for the data file:

Example

```
Router(config-bulk-tr)#buffersize 3072
```

The valid range is from 1024 to 2147483647 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.

3. Optional: Specify the format of the statistics file:

Example

```
Router(config-bulk-tr)#format schemaASCII
```

> **Note**
>
> Only `schemaASCII` is supported for transfers.

4. Associate one or more schemas with the transfer configuration:

Example

```
Router(config-bulk-tr)#schema TenGigE 0/5/0/11/1
```

5. Optional: Set how often the statistics file is transferred (in minutes).

Example

```
Router(config-bulk-tr)#transfer-interval 20
```

6. Define the primary transfer destination and protocol (URL).

Example

```
Router(config-bulk-tr)# url primary
ftp://user:password@host/folder/bulkstat1
```

7. Define a secondary (backup) transfer destination:

Example

```
Router(config-bulk-tr)# url secondary
tftp://10.1.0.1/tftpboot/user/bulkstat1
```

8. Set the number of transfer retries if the transfer fails:

   Example

   ```
   Router(config-bulk-tr)#retry 1
   ```

9. Set how long (in minutes) the statistics file is retained after transfer:

   Example

   ```
   Router(config-bulk-tr)#retain 60
   ```

   The valid range is from 0 to 20000. For retries, retain must be greater than 0.

10. Enable the bulk statistics transfer process.

    Example

    ```
    Router(config-bulk-tr)#enable
    ```

The device is now configured to periodically collect and transfer bulk statistics (MIB data) to the specified network management station. Data transfer will begin based on the defined interval and will follow your chosen file size, format, destinations, and retry options.

Configure bulk statistics transfer options

Configure options to determine how collected bulk statistics (MIB) data are transferred from the device to a remote management station.

Set up periodic transfer of collected bulk MIB data using SNMP bulkstat, ensuring efficient file management and reliable delivery to a network management system.

Before starting, ensure you have defined both bulk statistics object lists and bulk statistics schemas. These prerequisites must be configured to enable bulk statistics transfer. The following steps guide you through configuring the transfer options for periodic MIB data collection.

Before you begin

- Bulk statistics object lists must be defined.
- Bulk statistics schemas must be defined.

Procedure

1. Define the bulk statistics transfer configuration.

Example

```
Router# configure terminal
Router(config)# snmp-server mib bulkstat transfer bulkstat1
```

2. Optional: Set the maximum buffer size (in bytes) for the data file:

   Example

   ```
   Router(config-bulk-tr)#buffersize 3072
   ```

   The valid range is from 1024 to 2147483647 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.

3. Optional: Specify the format of the statistics file:

   Example

   ```
   Router(config-bulk-tr)#format schemaASCII
   ```

   > **Note**
   >
   > Only `schemaASCII` is supported for transfers.

4. Associate one or more schemas with the transfer configuration:

   Example

   ```
   Router(config-bulk-tr)#schema TenGigE 0/5/0/11/1
   ```

5. Optional: Set how often the statistics file is transferred (in minutes).

   Example

   ```
   Router(config-bulk-tr)#transfer-interval 20
   ```

6. Define the primary transfer destination and protocol (URL).

   Example

   ```
   Router(config-bulk-tr)# url primary
   ftp://user:password@host/folder/bulkstat1
   ```

7. Define a secondary (backup) transfer destination:

   Example

   ```
   Router(config-bulk-tr)# url secondary
   tftp://10.1.0.1/tftpboot/user/bulkstat1
   ```

8. Set the number of transfer retries if the transfer fails:

   Example

   ```
   Router(config-bulk-tr)#retry 1
   ```

9. Set how long (in minutes) the statistics file is retained after transfer:

   Example

   ```
   Router(config-bulk-tr)#retain 60
   ```

   The valid range is from 0 to 20000. For retries, retain must be greater than 0.

10. Enable the bulk statistics transfer process.

    Example

    ```
    Router(config-bulk-tr)#enable
    ```

The device is now configured to periodically collect and transfer bulk statistics (MIB data) to the specified network management station. Data transfer will begin based on the defined interval and will follow your chosen file size, format, destinations, and retry options.

Example of periodic MIB data collection and transfer

Provides sample configuration commands for periodic MIB data collection and transfer, along with example bulk statistics file

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
```

6 Configuration and File System Management

Topics:

- Secure file transfers
- Auto-save configurations
- Auto-save and copy router configuration using public key authentication
- Commit limits
- View VRF-specific configurations
- Monitor mount points

Describes advanced configuration management and file transfer enhancements, including secure transfer protocols, automated backup features, and commit limit adjustments for reliable system maintenance.

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

Secure file transfers

Explains how secure file transfers protect data on routers by using SFTP and SCP protocols for safe file movement between locations.

A secure file transfer is a router feature that

- duplicates files or data from one location to another,
- uses protocols like SFTP and SCP to safeguard data during transit, and
- retains the original content while enabling creation of backups or distribution copies.
- SFTP (Secure File Transfer Protocol): A protocol for secure, encrypted transfer of large files between systems.
- SCP (Secure Copy Protocol): A protocol for securely transferring files between servers using SSH.

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Secure file transfer from the Router	Release 7.9.1	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. This feature modifies the **copy** command.

Starting with Cisco IOS XR Release 7.9.1, secure file transfer supports SFTP and SCP through the router's copy command, enhancing data protection during transit. Public-key authentication allows users to securely transfer running configuration files to remote servers without specifying a password.

Copy the running configuration to a remote server using SFTP or SCP

Configure the router to securely transfer the running configuration file to a remote SFTP or SCP server.

Enable secure backup or migration of your router configuration files to a remote location.

Storing configuration files off-device helps with backup, compliance, and migration, and SFTP/SCP protocols provide encrypted transfer for security.

Before you begin

- Ensure SFTP or SCP access is available on the remote server.
- Confirm that the router's SSH server is enabled.
- Verify you have credentials for the remote server.

Procedure

1. Copy the running configuration file from the router to a remote server.

   - Using SFTP

```
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
```

   - Using SCP:

```
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
```

2. Verify the file on the SFTP or SCP server.

   - For SFTP

```
[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
```

   - For SCP

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

Example: Configure SCP and SFTP using public-key authentication

Explains how public-key authentication works with SCP and SFTP protocols

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
```

Auto-save configurations

Explains the auto-save configurations feature for Cisco routers, including its key attributes, functionality, and configurability.

A auto-save configuration is a router backup feature that

- automatically saves a backup of the running configuration to a specified location after each commit,
- supports storage both locally on the router and remotely via secure file transfer protocols, and
- provides flexible parameters for password encryption, file limits, wait-time, and timestamping backups.

The auto-save configurations feature enhances router reliability and disaster recovery by creating consistent, automated backups of all configuration changes. Once enabled, each committed change is saved without manual intervention, protecting against data loss and configuration errors.

Starting with Cisco IOS XR Software Release 7.10.1, auto-save supports public-key authentication for remote configuration backups.

Key configurable attributes:

- **scp and sftp protocols**: Save configuration backups to remote servers using Secure Copy Protocol (SCP) or Secure File Transfer Protocol (SFTP). SCP is a secure protocol for general file copying; SFTP is recommended for transferring large files.
- **password**: You can save encrypted passwords for both remote and local locations to secure backup file transfers.
- **maximum**: Define the maximum number of auto-saved files to retain. Once the limit is reached, older files are overwritten by newer ones. The default is 1, with a maximum limit of 4,294,967,295 files.
- **timestamp**: Append a timestamp to each auto-saved file name, using the router's configured time and timezone. Example:`test_123.autosave.1.ts.Tue_Jan_31_15-15-51_805_IST`
- **wait-time**: Specify the interval to wait before the next auto-save occurs (in days, months, or hours after a commit). Default is zero

Restrictions for auto-save configuration

Outlines the requirements and limitations for using the auto-save configuration.

Observe these restrictions to ensure correct operation of auto-save configuration.

- The maximum number of configuration lines currently supported on the Cisco 8000 Series routers is 1Million.
- Use only local paths, SCP, and SFTP paths for auto-save configuration; other paths are not supported.

Configure auto-save for configuration commits

Configure automatic saving of device configuration commits to prevent loss of configuration changes.

Enable auto-save for configuration commits so all changes are automatically preserved as you make modifications.

In environments where frequent configuration changes are made, configuring auto-save ensures all changes are recorded and reduces the risk of losing updates due to unexpected events.

Before you begin

- Identify the desired protocol, username, server, and file path for storing configuration backups.
- Prepare credentials if secure options (e.g., encrypted password) are needed.

Procedure

1. Enable auto-save for configuration commits:

   Example

   ```
   Router(config)#configuration commit auto-save
   ```

2. Optionally, specify the destination file for auto-saved configurations using the protocol, username, host, and file path:

   Example

   ```
   Router(config-cfg-autosave)#configuration commit auto-save filename
   sftp://user1@server1://test-folder/test_123
   ```

3. Set the password (clear or encrypted):

   Example

   ```
   Router(config-cfg-autosave)#password clear encryption-default cisco
   ```

4. Enable timestamping for saved files:

   Example

   ```
   Router(config-cfg-autosave)#timestamp
   ```

5. Set the maximum number of saved configurations.

   Example

   ```
   Router(config-cfg-autosave)#maximum 10
   ```

6. Configure the wait-time between successive auto-saves.

   Example

   ```
   Router(config-cfg-autosave)#wait-time days 0 hours 0 minutes 0 seconds 5
   Router(config-cfg-autosave)# commit
   ```

7. Verify the running configuration to ensure auto-save is enabled.

Example

```
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
!
```

All configuration commits are automatically saved based on the settings you specify, ensuring backups are properly maintained.

Auto-save and copy router configuration using public key authentication

Explains how auto-save and copy operations on router configurations use public key authentication to enable secure, passwordless operation and automation.

A router configuration auto-save and copy operation is a security feature that

- supports passwordless authentication for saving and copying running configurations,
- uses public key-based authentication to verify access securely, and
- automates configuration management and reduces password-related risks.

From Cisco IOS XR Software Release 7.10.1 onwards, you can use public key-based authentication without remembering or entering your password. This enables automatic saving of the running configuration and copying the configuration from a source (such as a network server) to a destination (such as a flash disk) in a secure, automated way. Using SSH with public key authentication prevents password disclosure and leakage, and supports single sign-on. The public key is mathematically linked to a private key; the private key is secret, while the public key is distributed on servers. The SSH server generates an authentication challenge using the public key, and only the private key can answer it, ensuring secure login.

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Auto-Save and Copy Router Configuration Using Public Key Authentication	Release 25.4.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on: • 8011-32Y8L2H2FH • 8011-12G12X4Y-A/D

Feature Name	Release Information	Feature Description
Auto-Save and Copy Router Configuration Using Public Key Authentication	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: • 8011-4G24Y4H-I • 8712-MOD-M
Auto-Save and Copy Router Configuration Using Public Key Authentication	Release 24.4.1	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*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
Auto-Save and Copy Router Configuration Using Public Key Authentication	Release 7.10.1	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. The feature introduces these changes: CLI: • The **configuration commit auto-save** command supports password-less authentication. • The **copy** command supports password-less authentication.

Configure auto-save using public key authentication

Configure auto-save functionality for secure connections using public key authentication.

Enable automatic saving of session data while maintaining security by authenticating with a public key.

Use auto-save with public key authentication to ensure session information is securely stored without manual intervention. This task is relevant when you need ongoing session storage while leveraging secure authentication methods.

Before you begin

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 **cyrpto 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 Chapter in System Security Configuration Guide for Cisco 8000 Series Routers.

Procedure

Configure the auto-save feature with an SFTP destination:

Example

```
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
```

The router will now automatically save its configuration to the specified SFTP server at the defined intervals using secure, passwordless public key authentication.

Commit limits

Explains how commit limits on Cisco IOS XR routers enable uninterrupted configuration changes by adjusting rebase and ASCII backup operations.

A commit limit is a configuration parameter on Cisco IOS XR routers that

- specifies how many configuration changes can be staged before triggering a backup operation,
- determines the maximum size of configuration data accumulated before a rebase or backup, and
- allows administrators to adjust thresholds to prevent commit interruptions during complex topology changes.

Cisco IOS XR routers use a two-stage configuration model:

- Target configuration stage: You build configurations using CLI commands, staging them before applying.

- Commit stage: The staged configuration is applied to the router using the `commit` command, updating the running configuration.

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.

Starting with Cisco IOS XR 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 cfgmgr 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.

Restrictions for increasing the commit limit

Outlines the restrictions and recommended actions when increasing the commit limit or managing configuration backups.

These restrictions apply when increasing the commit limit or managing backups:

- The **cfs check** command increases the rebase limits only for one instance. After the router performs a rebase operation, the limits reset to the default values of 20 commits and 2 MB configuration data. To enable the increased limits again, the **cfs check** command must be executed again.
- If the router reloads after executing the **cfs check** command, the rebase limits reset to the default 20 commits and 2 MB configuration data.
- When the router enters standby mode or reloads, the ASCII backup timer does not reset to zero; the router performs an ASCII backup 55 minutes after the first commit operation before standby or reload.
- Cisco does not recommend executing the **clear configuration inconsistency** and **clear configuration ascii inconsistency** commands regularly after each commit because these commands cause disk input/output operations that increase hard disk wear and tear.
- Frequent execution of these commands leads to frequent hard disk access, which can reduce the hardware lifespan.

Guidelines for increasing the commit limit

Lists commands for adjusting commit and configuration rebase limits on routers, and explains how ASCII backup and rebase operations behave.

Follow these guidelines for increasing the commit limit:

- Use the **cfs check** command to increase the rebase limits from the default 20 commits and 2 MB configuration data to 40 commits and 4 MB configuration data. This allows committing more changes without being blocked by rebase operations.
- After executing the **cfs check** command, the router will perform a rebase operation after 40 commits or when the configuration data reaches 4 MB.
- If the router switches over to standby RP after executing the **cfs check** command, the increased rebase limits (40 commits and 4 MB) are retained.

- Use the **clear configuration ascii inconsistency** command to initiate an ASCII backup and reset the ASCII backup timer to zero. This causes the router to perform the next ASCII backup only after 55 minutes from the command execution.
- The ASCII backup merges new commits into the ASCII backup file, and the timer must be reset by executing the **clear configuration ascii inconsistency** command again before the next 55 minutes if needed.
- These commands should be executed 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.

Increase the rebase limits

Configure the router to increase the commit count limit and configuration data limit in the Configuration File System.

Allow greater flexibility in commit operations by raising the commit count and configuration data limits on the router.

Increasing the rebase limits is useful when you need to maintain a larger commit history or handle larger configuration files, especially in environments where frequent changes are made to device configurations.

Before you begin

- Confirm there is sufficient memory available to accommodate increased limits.

Procedure

1. Confirm there is sufficient memory available to accommodate increased limits.

   Example

```
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 that the changes have been applied by viewing the configuration history.

   Example

```
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
```

The router now supports a higher commit count and larger configuration data size, allowing more configuration versions and larger configuratio

Perform ASCII backup and reset the ASCII backup timer

Configure your router to perform an immediate ASCII backup and reset the ASCII backup timer using CLI commands.

Backup startup and running router configuration in ASCII format and reset the ASCII backup timer count as a best practice for configuration management.

Performing an ASCII backup ensures a human-readable copy of your current configuration is saved. Resetting the ASCII backup timer may be required after resolving configuration inconsistencies.

-

Before you begin

- Ensure all nodes in the router are in IOS-XR RUN state.
- You may check the state of all nodes by running the `show platform` command.

Procedure

1. Verify that all nodes are in the RUN state.

   Example

   ```
   Router#show platform
   ```

   Confirm all nodes are ready before proceeding.

2. Perform the ASCII backup and reset the timer.

   Example

   ```
   Router#clear configuration ascii inconsistency
   Router# clear configuration ascii inconsistency
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Warning:
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   !!!! 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
   ```

   It is recommended to run this command only when all nodes in the router are in IOS-XR RUN state. To determine node state, run **`show platform`**.

   - When asked to proceed, confirm by entering `y`.
   - The router responds with progress and completion messages

     .

3. Optional: Verify that the backup and timer reset occurred.

   Example

   ```
   Router#show configuration history last 5
   ```

   Look for a **backup Periodic ASCII backup** event in the recent history.

Your router immediately performs an ASCII backup of the configuration and resets the ASCII backup timer to zero.

Concurrent configuration rebases

Explains how Cisco IOS XR routers enable simultaneous commit and rebase operations to prevent configuration interruptions during critical maintenance.

A concurrent configuration rebase is a router feature that

- allows commit and rebase operations to occur simultaneously,
- prevents configuration commits from being blocked during rebase operations, and
- enables complex topology changes without interruption.

Before Cisco IOS XR Release 24.3.1, rebase and ASCII backup operations blocked commit actions, requiring manual intervention such as increasing the maximum commit count or resetting the ASCII backup timer. Starting with Release 24.3.1, Cisco IOS XR routers allow concurrent commit and rebase operations, removing the need to use the `cfs check` command to adjust commit counts or diff sizes. However, ASCII backup operations may still block commits, and the ASCII backup timer can be reset using the `clear configuration ascii inconsistency` command to minimize downtime.

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Concurrent Configuration Rebase during Commit	Release 24.3.1	Introduced in this release on: Fixed Systems (8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) 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.

View VRF-specific configurations

Explains how VRF-specific configurations consolidate running configuration data and highlight logical dependencies within Cisco IOS XR.

A VRF-specific configuration is a network management feature that

- consolidates all running configuration data associated with a Virtual Routing and Forwarding (VRF) instance,
- supports visibility across various CLI link-points and keywords, and
- reveals logical parent-child dependencies for interfaces and bundles associated with a VRF.

Beginning with Cisco IOS XR Software Release 24.2.11, the `show running-configuration filter vrf` command enables you to filter configurations tied to a specific VRF. Previously, the `show running configuration` command listed configurations under distinct keywords only, which might not capture all related settings for a VRF.

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
View VRF-specific Configuration	Release 24.2.11	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. CLI: **show running-configuration filter vrf**

Restrictions for filtering running configuration for VRF

Outlines the limitations you must observe when filtering the running configuration for VRF-specific settings.

The VRF-specific configuration command has these limitations.

- 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
```

Monitor mount points

A mount-point provides access to additional file systems and is fundamental for managing storage resources.

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 9: Feature History Table

Feature Name	Release Information	Feature Description
Monitor mount points	Release 26.1.1	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]) 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. 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. This feature introduces these changes: **CLI:** • The **detail** keyword is added to the show filesystem command. **YANG Date Model:** • **openconfig-system:system/mount-points** Both Model-Driven Telemetry (MDT) and Event-Driven Telemetry (EDT) are supported. (See GitHub, YANG Data Models Navigator)

Restrictions for mount-point state monitoring

This section lists restrictions you must follow when performing operational reporting from mount points.

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.

7 Simple Network Management Protocol

Topics:

- SNMP
- Restrictions for SNMP usage
- SNMP versions
- SNMP security models and levels
- Security benefits of SNMPv3
- Comparison of response times for SNMP security models
- Security services of the SNMPv3 User-Based Security Model
- Access control in SNMP using the view-based access control model
- IP Precedence and DSCP options for SNMP traffic
- Custom MIB Support Using SNMP Operation Script
- Session MIB support on subscriber sessions
- SNMP notifications: traps versus informs
- Session types
- Specify SNMP notification parameters
- Configure the recipient of SNMP notifications
- Configure linkUp and linkDown traps for a subset of interfaces
- Configure persistent MIB data
- Configure SNMP trap notifications
- Configure SNMPv3
- Configure SNMP secure-v3
- Create a custom MIB using an SNMP script
- Set the maximum SNMP agent packet size
- Display the SNMP context mapping
- Display the SNMP context mapping table
- Enable BRIDGE-MIB trap notifications
- Monitor the packet loss
- Restrictions for custom MIB
- Set IP precedence and DSCP values
- Set the contact, location, and serial number of the SNMP agent

Describes the Simple Network Management Protocol (SNMP) framework, including security models, MIB management, notification parameters, and configuration tasks for effective network monitoring.

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.

- Configure the router to drop error PDUs

SNMP

Use this concept to understand the application-layer SNMP protocols and their components for network device monitoring and management.

A SNMP protocol is a network communication protocol suite that

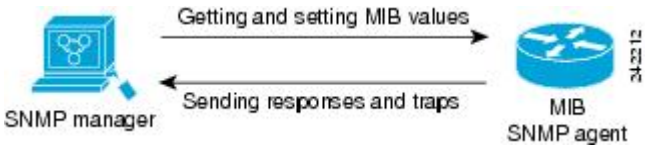
- provides a standardized framework for monitoring and managing network devices,
- defines roles for managers and agents, and
- enables structured data exchange using Management Information Bases (MIBs).

SNMP manager: The control system (often called a Network Management System, NMS) that sends requests to monitor or configure network devices.

SNMP agent: The software or firmware on managed devices that maintains local data and responds to requests from the SNMP manager, using information stored in the MIB.

Management Information Base (MIB): A structured repository of managed objects within a network device, grouped into modules and accessible through SNMP commands.

Figure 1: Communication between an SNMP agent and Manager



- SNMP is specified as an application-layer protocol for exchanging messages such as Get, Set, and Trap.
- Key MIB RFCs include STD 58, RFC 2578, RFC 2579, RFC 2580.
- IP-MIB (RFC4293) provides separate support for IPv4 and IPv6 statistics, with implementation details evolving across IOS-XR releases (from 6.3.2, both IPv4 and IPv6 statistics can be collected).
- Interface-specific statistics are indexed by address type (IPv4/IPv6) and interface ID (ifindex).
- Important OIDs added for IPv4/IPv6 support include: ipIfStatsInReceives, ipIfStatsHCInReceives, ipIfStatsInOctets, ipIfStatsOutTransmits, etc.
- For more details on supported MIBs, see the Cisco IOS XR MIBs tool and SNMP OID Navigator.

Restrictions for SNMP usage

This section lists the primary limitations and recommended usage guidelines for SNMP output and request handling

- SNMP outputs are only 32 bits wide, limiting display to values less than 4.29 Gigabits. Displaying interface speeds above this limit—such as 10 Gigabits—can cause concatenated or inaccurate results.
- The recommended maximum number of object identifiers (OIDs) in a single SNMP request is 75. Exceeding this may cause requests to be dropped or polling timeouts.

SNMP versions

This section explains SNMP versions, the different iterations of the Simple Network Management Protocol, and their major distinguishing attributes for network management and monitoring.

Starting with Cisco IOS XR Release 26.3.1,

- SNMP secure-v3 enforces authentication and encryption for every configured user. For more information, see Configure SNMP secure-v3 on page 85.
- Legacy SNMPv1, SNMPv2c, and SNMPv3 configurations are deprecated but remain supported. If a deprecated SNMP configuration remains active, the router logs a deprecation warning every 30 days.

A SNMP version is a network protocol specification that

- supports communication between network managers and managed devices,
- varies in security models and message handling, and
- introduces different feature sets such as bulk retrieval or enhanced authentication.

- SNMPv1: Operates with basic community-based security, providing read-only access using a shared password.
- SNMPv2c: Adds bulk retrieval for optimizing requests and provides more detailed error reporting while retaining community-based security.
- SNMPv3: Integrates a security model supporting authentication and privacy, allowing for stricter access control and standards compliance (RFCs 3411-3418).

SNMPv1 and SNMPv2c rely on defined communities for access control, usually controlled via IP address lists and passwords. SNMPv2c introduces mechanisms that reduce network traffic by allowing the retrieval of larger sets of data in fewer transactions, as well as extended error codes to aid troubleshooting.

SNMPv3 advances the protocol by providing configurable security levels, such as no authentication (noAuthNoPriv), authentication without privacy (authNoPriv), and authentication with privacy (authPriv), thereby allowing different security requirements per deployment.

Comparison of SNMPv1, SNMPv2c, and SNMPv3 feature support

This reference lists and compares the SNMP operations and features supported by SNMPv1, SNMPv2c, and SNMPv3.

SNMP Operations

- get-request: Retrieves a value from a specific variable.
- get-next-request: Retrieves the value of the variable that comes after the named variable, enabling sequential searching within a MIB table.
- get-response: Sends a response to a get-request, get-next-request, or set-request initiated by an NMS.
- set-request: Stores a value in a specific variable.
- trap: An SNMP agent sends an unsolicited message to an SNMP manager when an event occurs.

Table 10: SNMP Feature Support Comparison

Feature	SNMPv1	SNMPv2c	SNMPv3
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

Feature	SNMPv1	SNMPv2c	SNMPv3
Privacy (Encryption)	No	No	Yes
Authorization and Access Controls (Views)	No	No	Yes

**Not supported on Cisco IOS XR software

SNMP security models and levels

This section lists the SNMP security models and their supported security levels, showing the authentication and encryption mechanisms for each combination.

Starting with Cisco IOS XR Release 26.3.1,

- SNMP secure-v3 enforces authentication and encryption for every configured user. For more information, see Configure SNMP secure-v3 on page 85.
- Legacy SNMPv1, SNMPv2c, and SNMPv3 configurations are deprecated but remain supported. If a deprecated SNMP configuration remains active, the router logs a deprecation warning every 30 days.

Simple Network Management Protocol (SNMP) uses various security models and levels to control authentication and encryption for SNMP messages. Each security model (SNMPv1, SNMPv2c, SNMPv3) supports specific security levels, which determine if messages are authenticated and/or encrypted, impacting how SNMP messages are processed and protected.

Security Levels

- noAuthNoPriv: No authentication or privacy.
- authNoPriv: Authentication only, no privacy.
- authPriv: Authentication and privacy (encryption)

Table 11: SNMP Security Model and Level Combinations

SNMP Security 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.

SNMP Security Model	Level	Authentication	Encryption	What Happens
v3	authNoPriv	HMAC-MD5 or HMAC-SHA	No	Provides authentication based on the HMAC (Hash-Based Messsage Authentication Code)-MD5 (Message Digest 5) algorithm or the HMAC-SHA.
v3	authPriv (encryption-aes or encryption-default)	HMAC-MD5 or HMAC-SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES (Data Encryption Standard) 56-bit encryption in addition to authentication based on the CBC (Cipher Block Chaining) DES (DES-56) standard.
v3	authPriv (encryption-aes or encryption-default)	HMAC-MD5 or HMAC-SHA	3DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides 168-bit 3DES (Triple Data Encryption Standard) level of encryption.
v3	authPriv (encryption-aes or encryption-default)	HMAC-MD5 or HMAC-SHA	AES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides 128-bit AES (Advanced Encryption Standard) level of encryption.
secure-v3	authPriv (encryption-aes)	SHA-224, SHA-256, SHA-384, or SHA-512	AES-128, AES-192 and AES-256	Provides authentication based on the SHA-224, SHA-256, SHA-384, or SHA-512 algorithms. Provides 128-bit AES level of encryption.

Use of 3DES and AES encryption standards requires that the security package (k9sec) be installed. For information on installing software packages, see Chapter Upgrading and Managing Cisco IOS XR Software in System Setup and Software Installation Guide for ASR 9000 Series Routers.

Security benefits of SNMPv3

This section lists the key security features that SNMPv3 adds and the types of security threats it addresses.

Starting with Cisco IOS XR Release 26.3.1,

- SNMP secure-v3 enforces authentication and encryption for every configured user. For more information, see Configure SNMP secure-v3 on page 85.
- Legacy SNMPv1, SNMPv2c, and SNMPv3 configurations are deprecated but remain supported. If a deprecated SNMP configuration remains active, the router logs a deprecation warning every 30 days.

SNMPv3 provides secure access to devices through:

- authentication,
- encryption, and
- access control.

These enhancements protect against the following threats:

- Masquerade: An SNMP user assumes another's identity to perform unauthorized operations.
- Message stream modification: Messages are maliciously reordered, delayed, or replayed to cause unauthorized management operations.
- Disclosure: Packet exchanges are eavesdropped.

SNMPv3 also enforces access control over SNMP managed objects, further reducing the risk of unauthorized access.

Comparison of response times for SNMP security models

This table lists the order of SNMP response times for different security models and security levels.

SNMPv3 authentication and encryption introduce minor processing delays compared to SNMPv2c, but these are outweighed by security advantages. This table lists the response time order (from least to greatest) for each combination:

Table 12: Order of Response Times from Least to Greatest

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

Security services of the SNMPv3 User-Based Security Model

This section provides an overview of the security services and mechanisms provided by the SNMPv3 User-Based Security Model (USM) to protect network management operations.

The SNMPv3 User-Based Security Model includes these security services:

- Message integrity: Ensures messages are not altered or destroyed in an unauthorized way.
- Message origin authentication: Confirms the claimed identity of the originating user.
- Message confidentiality: Prevents disclosure of information to unauthorized persons.

SNMPv3 USM employs these protocols:

- Authentication: HMAC-MD5-96, HMAC-SHA-96
- Privacy: CBC-DES (DES-56) for encryption

Additional features:

- Only configured users can perform management operations.
- All SNMP messages are protected using authentication and privacy protocols.

Access control in SNMP using the view-based access control model

This section provides an overview of the View-Based Access Control Model (VACM), describing how it enables controlled SNMP access to managed objects using views and access policies.

View-Based Access Control Model (VACM):

The View-Based Access Control Model enables SNMP users to control access by assigning read, write, or notify privileges to SNMP managed objects. It prevents access to objects restricted by views. These access policies are set during user group configuration with the snmp-server group command.

MIB views:

- Restrict group access rights to specific management information within the domain for security.
- MIB views define managed object types and instances that can be viewed.

Access policy:

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

- Read-view access: Allows reading a set of authorized object instances.
- Write-view access: Allows writing to a set of authorized object instances.
- Notify-view access: Allows sending notifications regarding a set of authorized object instances.

IP Precedence and DSCP options for SNMP traffic

This section provides details on how IP Precedence and DSCP values affect quality of service for SNMP traffic on routers, including configuration options and technical definitions.

SNMP IP Precedence and differentiated services code point (DSCP) support deliver QoS specifically for SNMP traffic:

- You can set SNMP traffic generated by a router to use a specific QoS class, controlling its priority for network handling.
- The configured IP Precedence or DSCP value determines packet treatment in weighted random early detection (WRED).
- A router cannot assign different QoS classes to different SNMP traffic types—all SNMP traffic from the device shares the same configuration.

- Technical details:
  - IP Precedence: Refers to the first three bits of the type of service (ToS) byte in an IP header.
  - DSCP (Differentiated Services Code Point): Refers to the first six bits of the Differentiated Services (DiffServ) field byte.
  - Up to eight different IP Precedence values and 64 DSCP markings can be configured.

Custom MIB Support Using SNMP Operation Script

Use this concept to extend device monitoring by defining custom Management Information Bases (MIBs) that expose additional operational data through SNMP without requiring a software upgrade.

A custom MIB is a network management information base extension that

- allows creation of user-defined managed objects and their OIDs,
- retrieves real-time operational data using scripts triggered by SNMP requests, and
- enables support for monitoring requirements not covered by built-in MIBs.

With this feature, you can define a custom script for a given OID. When an SNMP request for the custom OID is received, the associated script retrieves data from the operational database and returns it to the Network Management System (NMS) using SNMP protocol.

Table 13: Feature History Table

Restrictions for custom MIB

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

Create a custom MIB using an SNMP script

Use this procedure to create and register a custom MIB object identifier (OID) with an SNMP script to read LLDP state information from your router.

Configure a custom MIB entry by mapping an SNMP script to a unique OID for LLDP state reporting.

This task enables SNMP management tools to access LLDP state information by defining and associating a custom OID with a Python script that retrieves LLDP operational data from the router.

Before you begin

- Prepare your SNMP script to query the LLDP status (example provided below).
- The custom OID must extend from the root: 1.3.6.1.4.1.9.9.999998.

Procedure

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

Example

```
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 this script.

```
Router#show_lldp_string.py
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. Generate the checksum of the script file.

Example

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

```
156345c2cbfc1a2725b5f5ecdfb23d30d9a25e894604890d88929d724946e7b3
show_lldp_string.py
```

5. Enter the configuration mode of the router.

Example

```
Router#configure
```

6. Enable read-write community string, where public is the read-write community.

Example

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

7. Map the script file to the custom OID using the **snmp-server script script-oid** *OID-number* **script-filename** *file-name* command.

Example

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

8. Configure the checksum of the script file using the **script snmp** *file-name* **checksum sha256** *checksum-value* command.

   Example

   ```
   Router(config)#script snmp show_lldp_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.

9. Verify that the SNMP has received get request for the custom OID.

   Example

   ```
   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

This section lists the supported features of CISCO-SUBSCRIBER-SESSION-MIB for monitoring subscriber session data via SNMP.

The CISCO-SUBSCRIBER-SESSION-MIB provides SNMP-based monitoring of per-subscriber and aggregate subscriber data such as PPPoE. It supports:

- Per-subscriber and aggregate session data collection,
- Notifications (traps) for aggregate session counts crossing pre-configured thresholds,
- Optimized data collection through integration with a generic MIB Data Collector Manager (DCM) for faster and more scalable reporting.

SNMP notifications: traps versus informs

This section lists the key differences between SNMP traps and informs, including their reliability, resource usage, and best-use scenarios.

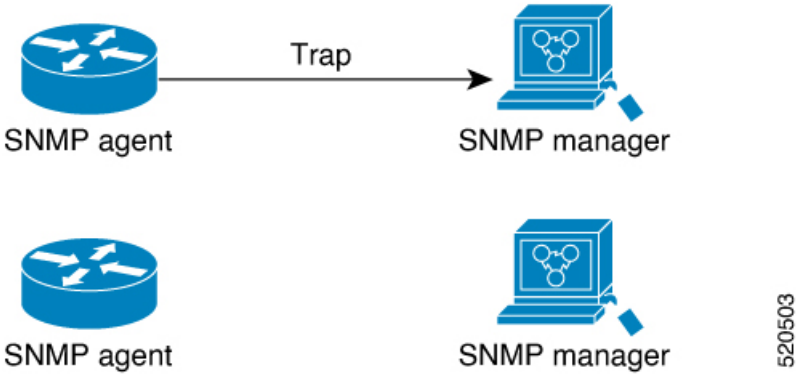
SNMP notifications are messages sent from an SNMP agent to an SNMP manager to alert about specific conditions or events on a network. The two main types of notifications are traps and informs, each with different reliability and resource requirements.

SNMP Traps

- Traps are asynchronous SNMP notifications sent by the agent to report significant events to the SNMP manager.
- They are less reliable because the manager does not acknowledge receipt, so the agent never knows if the notification arrived.
- Traps consume fewer resources since they are immediately discarded after being sent.
- They are suitable for non-critical alerts where resource efficiency is prioritized over guaranteed delivery.
- On Cisco IOS XR software, only traps are supported for unsolicited notifications.

> **Note**
>
> In recent Cisco IOS XR software releases, SNMP agents send traps to all unreachable hosts instead of dropping them after a timeout, increasing the likelihood of delivery.

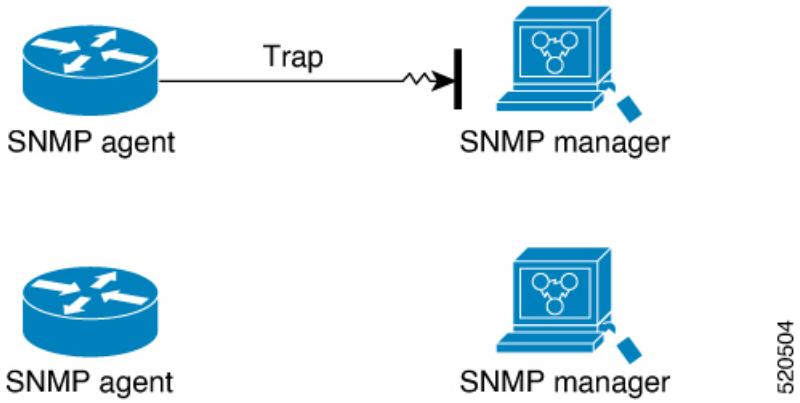
Figure 2: Traps received by SNMP Manager



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

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

Figure 3: Traps not received by SNMP manager



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.

SNMP Informs

- Informs are SNMP notifications that require acknowledgment from the SNMP manager.
- If the manager receives an inform, it responds with an SNMP response Protocol Data Unit (PDU). If not, the agent resends the inform until it's acknowledged or times out.
- Informs are more reliable but also more resource intensive, as the agent retains the inform in memory until acknowledged.
- They may be retried multiple times, increasing network traffic and overhead.
- Inform requests are supported from Cisco IOS XR Software Release 4.1 onwards.
- Informs are best for critical alerts where delivery assurance is essential.

Table 14: SNMP Notifications: Traps V/s Informs

Feature	Traps	Informs
Reliability	No acknowledgment; not retried	Require acknowledgment; retried if not received
Resource Usage	Minimal (immediately discarded after send)	Higher (held in memory until acknowledged or timed out)
Network Overhead	Low	Higher (possible multiple retries)
Use Case	Non-critical events, conserve resources	Critical events, delivery assurance required
Supported in IOS XR	Yes	Yes (from Release 4.1 onwards)

Session types

Describes the supported subscriber session types, including PPPoE, IP Subscriber Packet, and IP Subscriber DHCP, which can be monitored via SNMP to track connection status, performance, and subscriber data.

These are the supported session types:

- Point-to-Point Protocol over Ethernet (PPPoE) - PPPoE is a network protocol that encapsulates PPP frames inside Ethernet frames. It is commonly used for establishing internet connections over DSL lines. In SNMP, PPPoE sessions can be monitored to check the status, performance, and configuration of connections established using this protocol.
- IP Subscriber Packet (IP SUB PKT) - This typically refers to the handling of IP packets for subscribers in a network, often related to broadband or other IP-based services. Monitoring IP SUB PKT sessions via SNMP can involve tracking packet flow, session status, and subscriber-specific data to ensure proper service delivery and network performance.
- IP Subscriber DHCP (IP SUB DHCP) - This relates to IP address assignment using the DHCP (Dynamic Host Configuration Protocol) for network subscribers. DHCP automatically assigns IP addresses and other network configuration details to devices. SNMP can be used to monitor DHCP sessions, including lease status, address allocation, and overall DHCP server performance.

Specify SNMP notification parameters

Use this procedure to configure the SNMP notification source interface, message queue length, and retransmission interval to meet your network requirements.

Adjust SNMP notification parameters to optimize how your system manages and sends SNMP traps and notifications

This task helps you customize the SNMP notification source interface, message queue length, and retransmission interval for more precise network monitoring and alerting.

Before you begin

Ensure SNMP notifications are enabled.

Procedure

1. Optional: Specify the source interface for SNMP trap notifications.

   Example

   ```
   Router(config)#snmp-server trap-source POS 0/0/1/0
   ```

2. Optional: Set the message queue length for each SNMP host.

   Example

   ```
   Router(config)#snmp-server queue-length 20
   ```

3. Optional: Define the retransmission interval for notifications on the retransmission queue.

Example

```
Router(config)#snmp-server trap-timeout 20
```

The system applies your custom settings for SNMP trap source interface, message queue length, and retransmission interval.

What to do next

Configure the recipient of SNMP notifications

Use this procedure to configure an SNMP notification recipient (host) to receive SNMP traps from the device.

Enable SNMP notification delivery by specifying the host and parameters for receiving SNMP traps and notifications from your network device.

Starting with Cisco IOS XR Release 26.3.1,

- SNMP secure-v3 enforces authentication and encryption for every configured user. For more information, see Configure SNMP secure-v3 on page 85.
- Legacy SNMPv1, SNMPv2c, and SNMPv3 configurations are deprecated but remain supported. If a deprecated SNMP configuration remains active, the router logs a deprecation warning every 30 days.

Configuring the SNMP notification recipient allows the network device to send event notifications (traps or informs) to a designated SNMP manager for monitoring and management.

Before you begin

- Ensure SNMP is enabled on the device.
- Obtain the IP address of the SNMP manager (recipient).
- Determine the SNMP version and community string to be used.

Procedure

Specify the recipient host and notification parameters.

Example

```
Router(config)#snmp-server host 223.255.254.249 traps version 2c cebridge1
udp-port
```

Table 15:

If...	Then...
You want to specify a custom port	Include `udp-port <port-number>` in the command.
Using SNMP version 3	Supply the correct v3 user and security parameters

If...	Then...
SNMP manager IP changes	Update the host IP in the `snmp-server host` command.

Configure linkUp and linkDown traps for a subset of interfaces

Use this task to enable or disable SNMP linkUp and linkDown traps for a specific group of interfaces by specifying a regular expression. This approach streamlines monitoring and reduces unnecessary notifications at scale.

Enable or disable SNMP linkUp and linkDown interface notifications for only the interfaces you specify.

In large networks, enabling SNMP linkUp/linkDown traps for all interfaces can generate excessive notifications. Defining interface subsets lets you focus monitoring and reduce alert noise.

Before you begin

SNMP must be configured.

Procedure

1. Define an interface subset using a regular expression.

   Example

   ```
   Router(config)#snmp-server interface subset 10 regular-expression
   "^Gig[a-zA-Z]+[0-9/]+\."
   ```

   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.

2. Disable or enable linkUp and linkDown traps for all interfaces.

   Example

   ```
   Router(config-snmp-if-subset)#notification linkupdown disable
   ```

   To enable previously disabled interfaces, use the **no** form of this command.

3. Optional: Display the linkUp and linkDown notification status for all interfaces identified by the subset priority.

   Example

   ```
   Router#show snmp interface notification subset 10
   ```

4. Optional: Display the linkUp and linkDown notification status for all interfaces identified by the regular expression.

Example

```
Router#show snmp interface notification regular-expression
"^Gig[a-zA-Z]+[0-9/]+\."
```

5. Optional: Displays the linkUp and linkDown notification status for the specified interface.

Example

```
Router#show snmp interface notification tengige 0/4/0/3.10
```

The device enables or disables SNMP linkUp and linkDown traps only for the specified group of interfaces, helping you streamline monitoring and avoid unnecessary notifications.

Configure persistent MIB data

Use this task to enable persistent storage for SNMP MIB indices and cache QoS MIB data across process restarts, switchovers, and device reloads.

Ensure key SNMP MIB data and indices retain their values across process interruptions, improving management reliability and reducing re-indexing overhead.

In many SNMP MIB tables, indices serve as unique identifiers and are often mapped to internal data structures. For instance, ENTITY-MIB and CISCO-CLASS-BASED-QOS-MIB typically require persistent indices to avoid increased processing during network management tasks. Caching statistics for QoS MIBs helps optimize query response times and CPU utilization.

Before you begin

Identify which MIB tables require persistent indexing and caching in your deployment scenario.

Procedure

1. Optional: Enable persistent storage of the CISCO-CLASS-BASED-QOS-MIB data.

Example

```
Router#configure
Router(config)#snmp-server mibs cbqosmib persist
```

2. Optional: Enable QoS MIB caching with a specified cache refresh time.

Example

```
Router(config)#snmp-server mibs cbqosmib cache
refresh time 45
```

3. Optional: Enable QoS MIB caching with a limited number of service policies to cache.

Example

```
Router(config)#snmp-server mibs cbqosmib cache
service-policy count 50
```

4. Enables ifIndex persistence globally on all Simple Network Management Protocol (SNMP) interfaces.

   Example

   ```
   Router(config)#snmp-server ifindex persist
   ```

MIB index values and QoS MIB statistics persist through process restarts, switchovers, and device reloads. This ensures consistent network monitoring and reduces discovery time for SNMP management systems.

Configure SNMP trap notifications

Learn how to configure your router to send SNMP trap notifications using SNMPv3 for enhanced monitoring and security.

Enable SNMP trap notifications to inform your NMS of important events, using SNMPv3 for enhanced security.

SNMPv3 provides authentication and privacy for trap messages. This task describes how to configure user/groups and enable SNMP trap notifications.

Before you begin

Procedure

1. Configure a new SNMP group or a table that maps SNMP users to SNMP views.

   Example

   ```
   Router#configure
   Router(config)#router# snmp-server group group_name v3 noauth read view_name1
    writer view_name2
   ```

2. Create an SNMPv3 user and assign it to the group.

   Example

   ```
   Router(config)#snmp-server group group_name v3 noauth read view_name1 writer
    view_name2
   ```

3. Configure the SNMP trap recipient, version, and security level.

   Example

   ```
   Router(config)#snmp-server user noauthuser group_name v3
   ```

4. Specify SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the notifications.

Example

```
Router(config)#snmp-server host 12.26.25.61 traps version 3
noauth userV3noauth
```

5. Enable the sending of trap notifications and specifies the type of trap notifications to be sent.

Example

```
Router(config)#snmp-server traps bgp
```

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.

What to do next

Configure SNMPv3

Use this procedure to configure SNMPv3 for secure and efficient network management and monitoring on Cisco devices.

Use this task to enable and configure SNMPv3 on your device.

Starting with Cisco IOS XR Release 26.3.1,

- SNMP secure-v3 enforces authentication and encryption for every configured user. For more information, see Configure SNMP secure-v3 on page 85.
- Legacy SNMPv1, SNMPv2c, and SNMPv3 configurations are deprecated but remain supported. If a deprecated SNMP configuration remains active, the router logs a deprecation warning every 30 days.

SNMPv3 provides network monitoring and management with enhanced security features such as authentication and encryption.

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

1. Optional: Specify the identification number of local SNMP engine.

Example

```
Router#configure
Router(config)# 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.

2. Configure a new SNMP group or a table that maps SNMP users to SNMP views.

   Example

   ```
   Router(config)#snmp-server group
   group_name v3 noauth read view_name1 write view_name2
   ```

3. Configure a new user to SNMP group.

   Example

   ```
   Router#snmp-server user noauthuser group_name v3
   ```

4. Optional: Display information about the status of SNMP.

   Example

   ```
   Router#show snmp
   ```

5. Optional: Display information about the local SNMP engine.

   Example

   ```
   Router#show snmp group
   ```

6. Optional: Displays information about each SNMP username in the SNMP users table.

   Example

   ```
   Router#show snmp users
   ```

7. Optional: Display information about the configured views, including the associated MIB view family name, storage type, and status.

   Example

   ```
   Router#show snmp view
   ```

Configure SNMP secure-v3

Describes how SNMP secure-v3 CLI features enforce the secure network management by restricting configuration to strong authentication and encryption algorithms.

A SNMP secure-v3 feature is a network management configuration option that

- enables the secure-v3 keyword for SNMP CLI commands to require only SHA-2 authentication algorithms and AES-based privacy encryption,
- removes support for legacy SNMP versions (v1, v2c) and insecure cryptographic options such as MD5, SHA-1, DES, and 3DES, and

- requires explicit authentication and privacy settings for all SNMP users, with all passwords entered using `clear` or `encrypted` options and stored using AES encryption.

Table 16: Feature History Table

Feature Name	Release Information	Feature Description
Configure SNMP secure-v3	Release 26.3.1	Introduced in this release on: Fixed Systems-Cisco 8010 Series Routers (ASIC: A100), Cisco 8200 Series Routers (ASICs: Q100, Q200, and P100), and Cisco 8700 Series Routers (ASICs: P100 and K100). This feature enhances the SNMP security by configuring only SNMPv3 with SHA-2 authentication and AES-based privacy using the secure-v3 CLI. The router continues to support SNMPv1, SNMPv2c, and legacy SNMPv3 configurations, but these configurations are deprecated. If a legacy SNMP configuration remains active, the router logs a deprecation warning every 30 days. The router rejects the commit if the same group, user, or notification host user is configured through both legacy SNMP and SNMP secure-v3.

Restrictions for SNMP secure-v3

Describes the configuration and security restrictions that apply to SNMP secure-v3.

A SNMP secure-v3 requires authentication and privacy and supports only approved authentication and encryption algorithms.

- The same group or user cannot use both legacy and secure configuration.
- SNMP secure-v3 always uses the `authPriv` security level. The vsecure-v3 group, host, and VRF-specific host commands do not require a separate security-level keyword. The `secure-v3` user command requires the `auth` and `priv` keywords to configure the authentication and privacy algorithms and passwords.
- Only SHA-2 variants (such as SHA-224, SHA-256, SHA-384, and SHA-512) are supported for authentication. The MD5 and SHA-1 are not supported.
- SNMP secure-v3 supports AES-128, AES-192 and AES-256. DES and 3DES are not supported.
- A notification host-user can be associated with only one SNMP version.
- Legacy SNMP user and notification host configurations support type 6 or type 7 encryption. SNMP secure-v3 supports only type 6 encryption for these configurations.

Configure SNMP secure-v3

Configure SNMP secure-v3 CLI to enforce secure network management using strong authentication and encryption.

Create a secure SNMPv3 group and user. You can also configure trap or inform recipients in the default VRF or a specific VRF.

Secure SNMPv3 requires the `authPriv` security level. It supports SHA-224, SHA-256, SHA-384, or SHA-512 authentication and AES-128, AES-192 and AES-256 privacy.

Before you begin

- Remove all legacy SNMP v1, v2c, and v3 user, group, and host configuration from the device.
- Access privileged EXEC mode and global configuration mode.
- Enable AES password encryption before configuring the primary encryption key.

```
Router# configure
Router(config)# password6 encryption aes
Router(config)# commit
Router(config)# end
Router# key config-key password-encryption
Router# show type6 server
```

Follow these steps to configure SNMP secure-v3 CLI for user, group, and host setup.

Procedure

1. Enter global configuration mode.

   Example

   ```
   Router# configure
   ```

2. Create the SNMP secure-v3 group.

   Example

   ```
   Router(config)# snmp-server group <group-name> secure-v3
   ```

3. Create the SNMP secure-v3 user.

   Example

   ```
   Router(config)# snmp-server user <user-name> <group-name> secure-v3 auth
   {sha-224 | sha-256 | sha-384 | sha-512} {clear | encrypted} encryption-aes
   <authentication-password> priv aes {128 | 192 | 256} {clear | encrypted}
   encryption-aes <privacy-password>
   ```

   If a supplied password is already encrypted, replace the corresponding `clear` keyword with `encrypted`.

   The authentication-password and privacy-password choices are independent.

4. Configure a destination SNMP host for traps or informs using the secure-v3 version and user.

   Example

   ```
   Router(config)# snmp-server host <host-ip-address> version secure-v3
   {<user-name> | clear encryption-aes <user-name> | encrypted encryption-aes
   <encrypted-user-name>} [udp-port <port-number>]
   Router(config)# snmp-server host <host-ip-address> informs version secure-v3
    {<user-name> | clear encryption-aes <user-name> | encrypted encryption-aes
   <encrypted-user-name>} [udp-port <port-number>]
   Router(config)# snmp-server vrf <vrf-name> host <host-ip-address> version
   secure-v3 {<user-name> | clear encryption-aes <user-name> | encrypted
   encryption-aes <encrypted-user-name>} [udp-port <port-number>]
   Router(config)# snmp-server vrf <vrf-name> host <host-ip-address> informs
   ```

```
version secure-v3 {<user-name> | clear encryption-aes <user-name> | encrypted
 encryption-aes <encrypted-user-name>} [udp-port <port-number>]
```

- Use `<user-name>` to enter the username directly as a WORD value. Do not enter the `encryption-aes` keyword with this form.
- Use `clear encryption-aes <user-name>` to enter an unencrypted username and store it using AES encryption.
- Use `encrypted encryption-aes <encrypted-user-name>` when the supplied username is already encrypted.
- To specify a UDP port, include `udp-port <port-number>`.
- Enter only the command corresponding to the required trap or inform recipient and VRF.

5. Commit the configuration.

Example

```
Router(config)# commit
```

6. Verify the SNMP secure-v3.

Example

```
Router# show running-config snmp-server
Router# show snmp group
Router# show snmp user
Router# show snmp host
Router# show snmp trace configuration
```

The secure SNMPv3 group and user are created after the configuration is committed. If notification recipients are configured, the router sends secure SNMPv3 traps or informs to the specified destinations.

What to do next

- Confirm that the SNMP manager can authenticate the user and decrypt SNMP messages.
- To delete the configuration, see Remove SNMP secure-v3 configuration on page 88.

Remove SNMP secure-v3 configuration

Remove a secure SNMPv3 configuration when the group, user, or associated notification recipients are no longer required.

Remove SNMP secure-v3 notification recipients, users, and groups from the router.

Procedure

---

1. Remove a trap and inform recipients from the default VRF.

Example

```
Router# configure
Router(config)# no snmp-server host <host-ip-address> traps version secure-v3
 <user-name>
Router(config)# no snmp-server host <host-ip-address> informs version secure-v3
 <user-name>
```

2. Remove a trap and inform recipients from a specific VRF.

Example

```
Router(config)# no snmp-server vrf <vrf-name> host <host-ip-address> traps
version secure-v3 <user-name>
Router(config)# no snmp-server vrf <vrf-name> host <host-ip-address> informs
 version secure-v3 <user-name>
```

3. Remove a secure-v3 SNMP user and group if no longer is needed.

Example

```
Router(config)# no snmp-server user <user-name>
Router(config)# no snmp-server group <group-name>
```

4. Commit the changes.

Example

```
Router(config)# commit
```

After the commit, the specified SNMP secure-v3 group, user, and notification recipients are removed.

What to do next

Use the following commands to confirm the removal:

```
Router# show running-config snmp-server
Router# show snmp group
Router# show snmp user
Router# show snmp host
```

Create a custom MIB using an SNMP script

Use this procedure to create and register a custom MIB object identifier (OID) with an SNMP script to read LLDP state information from your router.

Configure a custom MIB entry by mapping an SNMP script to a unique OID for LLDP state reporting.

This task enables SNMP management tools to access LLDP state information by defining and associating a custom OID with a Python script that retrieves LLDP operational data from the router.

Before you begin

- Prepare your SNMP script to query the LLDP status (example provided below).

- The custom OID must extend from the root: 1.3.6.1.4.1.9.9.999998.

Procedure

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

   Example

```
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 this script.

```
Router#show_lldp_string.py
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. Generate the checksum of the script file.

   Example

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

```
156345c2cbfc1a2725b5f5ecdfb23d30d9a25e894604890d88929d724946e7b3
show_lldp_string.py
```

5. Enter the configuration mode of the router.

   Example

```
Router#configure
```

6. Enable read-write community string, where public is the read-write community.

   Example

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

7. Map the script file to the custom OID using the **snmp-server script script-oid** *OID-number* **script-filename** *file-name* command.

   Example

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

8. Configure the checksum of the script file using the **script snmp** *file-name* **checksum sha256** *checksum-value* command.

   Example

   ```
   Router(config)#script snmp show_lldp_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.

9. Verify that the SNMP has received get request for the custom OID.

   Example

   ```
   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
   ```

Set the maximum SNMP agent packet size

Use this procedure to configure the maximum SNMP packet size accepted or generated by the SNMP server.

Ensure that SNMP communication accommodates larger messages as needed for your environment by increasing the maximum packet size.

By default, SNMP packets may be limited in size. Configuring the packet size can help avoid issues with large SNMP responses or requests

Before you begin

Ensure SNMP is already enabled if you intend to adjust its parameters.

Procedure

Optional: Set the maximum packet size.

Example

```
Router#configure
Router(config)#snmp-server packetsize 1024
```

This sets the maximum packet size to 1024 bytes.

The SNMP server will now use the configured maximum packet size for incoming requests and outgoing replies.

Display the SNMP context mapping

Learn to display SNMP context mappings and understand the relationship between community names and context names on the router.

To examine the SNMP context mappings configured on your device, retrieve and review them.

SNMP context mappings link SNMP community names with logical network segments or features. This association allows more granular access and management within SNMP operations.

Before you begin

Procedure

1. Display the current SNMP context mappings.

   Example

   ```
   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
   ```

   The output lists the context-name, feature-name, and associated feature for each mapping.

2. Optional: View the full SNMP server configuration.

   Example

   ```
   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
   ```

   This will display all SNMP-related configuration including community strings, contexts, and community maps.

   In the command output, the **community name** identifies the SNMP community, and the **context name** specifies the logical SNMP context.

What to do next

Display the SNMP context mapping table

Use this task to display the current SNMP context mappings configured on your device.

View all SNMP context mappings to determine which client features and instances have created contexts.

SNMP context mapping allows features to create unique SNMP contexts for different operations. You may need to display this mapping for troubleshooting or audit purposes.

Procedure

Display the SNMP context mapping table.

Example

```
Router#show snmp context-mapping
```

The SNMP context mapping table displays, showing each context name, feature, and instance that created the context.

Enable BRIDGE-MIB trap notifications

Learn how to enable BRIDGE-MIB trap notifications on your router to send SNMP events related to network bridges to your SNMP management system.

Enable SNMP management systems to detect and respond to bridge-related events on your router through BRIDGE-MIB trap notifications.

Enabling BRIDGE-MIB trap notifications allows network monitoring tools to receive updates about changes or issues with network bridges, helping with faster troubleshooting and network stability.

Before you begin

SNMP is configured on your router.

Procedure

Enable BRIDGE-MIB trap notifications.

Example

```
Router#configure
Router(config)#snmp-server traps bridgemib
Router(config)#commit
```

Monitor the packet loss

Use SNMP traps to monitor packet loss when it exceeds defined thresholds.

Configure SNMP EVENT-MIB traps to automate monitoring of packet loss events if thresholds are breached.

This procedure helps you detect and respond to packet loss on interfaces, as SNMP notifications are generated automatically for network monitoring systems.

Before you begin

- Ensure SNMP is enabled and properly configured on your device.
- Confirm that your SNMP management system can receive and process EVENT-MIB traps.

Procedure

Configure SNMP to monitor packet loss by specifying the falling (lower) threshold, sampling interval, and rising (upper) threshold:

Example

```
Router#configure
Router(config)#snmp-server mibs eventmib packet-loss falling 1 interval 5
rising 2
Router(config)#end
```

Note

- `<lower-threshold>`: Minimum packet loss percentage to trigger a falling trap.
- `<sampling-interval>`: How often (in minutes, 5-1440) packet loss is polled.
- `<upper-threshold>`: Packet loss percentage to trigger a rising trap.
- Up to 100 interfaces can be monitored.

The device generates SNMP EVENT-MIB traps when packet loss on configured interfaces exceeds the upper threshold or drops below the lower threshold. These entries cannot be modified with SNMP SET operations.

Restrictions for custom MIB

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

Set IP precedence and DSCP values

Learn how to configure IP Precedence or DSCP values for SNMP traffic using the CLI.

Define the precedence or DSCP marking for SNMP traffic to enable prioritization by QoS mechanisms.

Use these commands to ensure SNMP packets are marked for differentiated treatment in the network.

Before you begin

SNMP must be configured.

Procedure

Configure an IPv4 precedence or IPv4 DSCP value for SNMP traffic. Use one of these commands:

- **snmp-server ipv4 precedence** *value*
- **snmp-server ipv4 dscp** *value*

Example

```
Router#configure terminal
Router(config)#snmp-server dscp 24
```

Similarly, you can use **ipv6** instead of **ipv4**to configure IP precedence and DSCP values for IPv6.

SNMP traffic now carries the specified IP Precedence or DSCP value.

What to do next

-

Set the contact, location, and serial number of the SNMP agent

Use this task to configure the system contact string, system location string, and system serial number for the SNMP agent.

Assign identifying information to the SNMP agent so network administrators can easily contact key personnel, locate devices, and track hardware accurately.

SNMP configuration improves device manageability and accountability within a networked environment.

Procedure

1. Optional: Set the system contact string.

   Example

   ```
   Router#configure
   Router(config)#snmp-server contact
   Dial System Operator at beeper # 27345
   ```

2. Optional: Set the system location string.

   Example

   ```
   Router(config)#snmp-server location
   Building 3/Room 214
   ```

3. Optional: Set the system serial number.

   Example

   ```
   Router(config)# snmp-server chassis-id 1234456
   ```

Configure the router to drop error PDUs

Learn how to configure the router so that it drops error PDUs when polled with an incorrect SNMPv3 user name.

Ensure the router does not send error PDUs for unknown SNMPv3 user names by configuring it to drop such requests.

This configuration helps 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.

Before you begin

Procedure

Configure the router to drop error PDUs when polled with an incorrect SNMPv3 user name.

Example

```
Router#configure
Router(config)#snmp-server drop unknown-user
```

8 Process Memory Management

Topics:

- Core dump folder limits

Describes memory management processes in Cisco IOS XR, focusing on diagnostic storage, core dump folder limits, and system reliability through efficient disk usage monitoring.

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.

Note

Starting with Cisco IOS XR Release 26.2.2, PAM is disabled in the 64-bit ARM architecture (8000-aarch64). As a result, the memory monitoring commands including **show memory-snapshots**, **show memory-top-consumers**, **show memory-delta**, and so on are not supported and do not display any output.

Core dump folder limits

Explains how configurable core dump folder limits in Cisco IOS XR control diagnostic file storage, automate cleanup, and help safeguard system reliability.

A core dump folder limit is a storage management feature that

- enables administrators to control the amount of disk space used for storing core dump diagnostic files,
- enforces the maximum number of coreinfo files retained in the core dump directory, and
- automates periodic cleanup and disk usage monitoring to prevent data loss during system failures.

Before Cisco IOS XR Release 25.2.1, the core dump folder lacked a storage limit, which meant repeated system crashes could fill all allocated disk space, preventing newer core files from being saved and resulting in potential data loss.

From Release 25.2.1, Cisco IOS XR supports the following mechanisms to manage core dump folder usage:

- Disk usage limit: Use the `exception disk-usage-limit <usage limit percent>` command to set core dumps to fill only 20% to 50% of the disk space.
- Throttling mechanism bypass: Use the `exception filepath` command to redirect core files to another directory, enabling new storage locations.
- Periodic cleanup: The system automatically checks the core dump folder every 15 minutes and deletes the oldest coreinfo files if their count exceeds 50, focusing on files like those ending with `.core.txt`.
- Syslog alerts: Syslog messages notify administrators when limits are reached, or automatic cleanup occurs, maintaining disk health and alerting to critical storage events.

Restrictions of core dump folder limit

Outlines the requirements for core dump storage management in Cisco IOS XR, including actions when disk space or folder limits are exceeded.

To ensure proper handling of core dumps in Cisco IOS XR, adhere to the following requirements:

- If available disk space is less than 10 percent of the hard drive, the system aborts the core file copy and deletes the file from the temporary directory.
- If the core dump folder exceeds the configured disk usage threshold and the most recent core file is within 10 minutes of the latest system crash, the system aborts saving new core files and deletes incoming files.

Examples:

- When available disk space is below 10 percent:
  - The system aborts the core file copy and deletes the core file from /misc/scratch/core.

    Example syslog message:

```
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
```

- When disk usage exceeds the folder limit and prior core dump is within 10 minutes of current crash:

- The system aborts the core file copy and deletes the core file from /misc/scratch/core.
- Example syslog message:

```
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

Describes how Cisco IOS XR software manages core dump storage by monitoring disk usage and enforcing folder limits, including how it responds to various disk usage scenarios.

Summary

Cisco IOS XR software manages core dump storage by monitoring disk usage in the core dump folder and enforcing configured folder limits. When certain disk usage thresholds are reached, the software deletes older core files or aborts saving new files to maintain storage integrity and prevent failures.

- The key components involved in the process are:
  - Cisco IOS XR software: Monitors disk usage and manages core file storage based on configured limits.
  - Core dump folder: Stores the core files created during system crashes.
  - Core files: Files generated during system crashes for troubleshooting and analysis.

Workflow

The process involves these stages:

1. Continuous monitoring: Cisco IOS XR software continuously monitors the disk usage of the core dump folder against the configured limit.
2. Exceeding disk usage limit and files older than 10 minutes:
   - If the core dump folder storage size exceeds the configured disk usage limit and the latest core file is more than 10 minutes older than the current crash time, the software deletes older core files (starting from oldest to newest) until storage usage falls below the configured limit.

     Example syslog messages:

```
Jan 21 10:19:36.251 UTC: dumper[399]: %OS-COREHELPER-6-DELETE_STALE_CORE
:
              Deleting /misc/disk1/coredumps/
sleep_10432.by.3.20250121-100007 ... Freeing up old
              core in /misc/disk1/coredumps, attempting to copy new core
file
```

3. Exceeding disk usage limit and files within 10 minutes of crash time:
   - If storage exceeds the configured limit but the latest core file is within 10 minutes of the current crash time, the software aborts saving new core files and deletes any incoming files.

Example syslog messages:

```
Jan 24 10:23:53.921 UTC: dumper[175]: %OS-COREHELPER-2-ABORT_COPY : Copy
of core
              file aborted for reason: Dumper disk usage exceeds 20%
threshold of /misc/disk1.
              Potential usage with new core: 100.0%
```

4. Hard disk usage exceeds 90%:
   - If the overall hard disk is filled more than 90%, the software aborts saving and deletes any incoming core files, as it cannot delete any old files to free up space.

     Example syslog messages:

```
Mar 25 04:25:06.252 UTC: dumper[329]: %OS-COREHELPER-2-ABORT_COPY : Aborted
 copy
              due to insufficient available space; deleting incoming core
 file.
```

Result

Cisco IOS XR software ensures that core dump storage does not exceed configured limits, maintaining system stability and preventing disk overflow. Old files are deleted or new files are blocked when necessary, with syslog messages generated for each action.

Configure core dump folder limit

Configure the storage limit for the core dump folder to prevent excessive consumption of hard disk space.

Limit the percentage of hard disk space used for core dumps to maintain available storage and system performance.

By controlling the storage allocation for core dumps, you prevent the system from using excessive disk space, which can affect system operation.

Procedure

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.

2. Verify the configured 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%
```

The router restricts core dump storage usage to the configured percentage of total disk space, preventing excessive disk consumption.

What to do next

-