System Management Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 24.1.x , 24.2.x , 24.3.x

First Published: 2023-11-30

Americas Headquarters

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

Preface

From Release 6.1.2 onwards, Cisco introduces support for the 64-bit Linux-based IOS XR operating system. Extensive feature parity is maintained between the 32-bit and 64-bit environments. Unless explicitly marked otherwise, the contents of this document are applicable for both the environments. For more details on Cisco IOS XR 64 bit, refer to the Release Notes for Cisco ASR 9000 Series Routers, Release 6.1.2 document.

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

Changes to This Document

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

Table 1: Changes to This Document

Date	Summary
June 2024	Republished for Release 24.2.1
February 2024	Initial release of this document

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

CHAPTER 1

New and Changed System Management Features

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

System Management Features Added or Modified in IOS XR Release 24.X.X

Feature	Description	Changed in Release	Where Documented
Performance Monitoring for PTP Networks	This feature is introduced.	Release 24.3.1	Performance Monitoring for PTP Networks, on page 400
Isolate Foreign Masters Causing Packet Timing Signal Fail	This feature is introduced.	Release 24.2.1	Isolate Foreign Masters Causing Packet Timing Signal Fail, on page 366
PTP Phase Difference Between Passive and Slave Ports	This feature is introduced.	Release 24.2.1	PTP Phase Difference Threshold Between Passive and Secondary Ports, on page 363

CHAPTER 2

YANG Data Models for System Management Features

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

Using YANG Data Models

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

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

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

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

CHAPTER 3

Configuring Simple Network Management Protocol

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

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

For detailed conceptual information about SNMP on the Cisco IOS XR software and complete descriptions of the SNMP commands listed in this module, see Related Documents, on page 33. For information on specific MIBs, refer to *Cisco ASR 9000 Series Aggregation Services Routers MIB Specifications Guide*. To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 2: Feature History for Implementing SNMP on Cisco IOS XR Software

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.9.0	Support was added for 3DES and AES encryption. The ability to preserve ENTITY-MIB and CISCO-CLASS-BASED-QOS-MIB data was added.
Release 4.2.0	Support was added for SNMP over IPv6.

This module contains the following topics:

Prerequisites for Implementing SNMP

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

Restrictions for SNMP Use on Cisco IOS XR Software

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

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

Information About Implementing SNMP

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

SNMP Functional Overview

The SNMP framework consists of three parts:

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

SNMP Manager

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

SNMP Agent

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

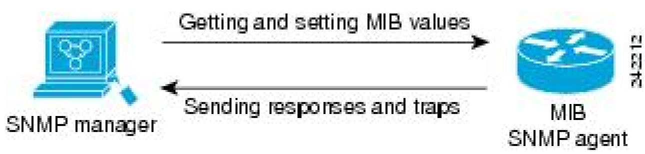
MIB

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

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

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

Figure 1: Communication Between an SNMP Agent and Manager



IP-MIB Support

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

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

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

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

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

Related Topics

Additional References, on page 33

SNMP Notifications

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

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

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

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

Figure 2: Trap Received by the SNMP Manager

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

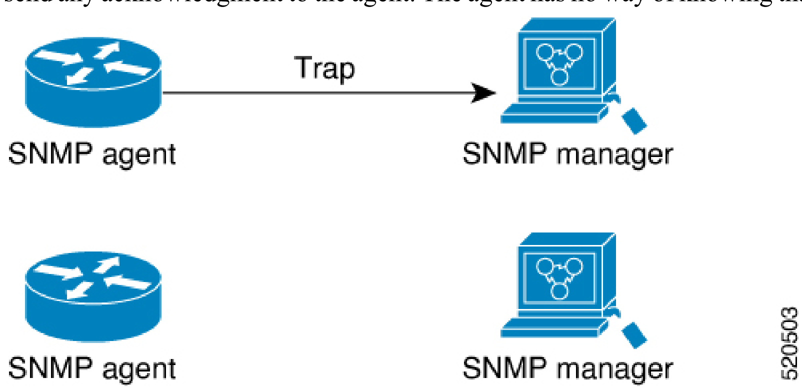
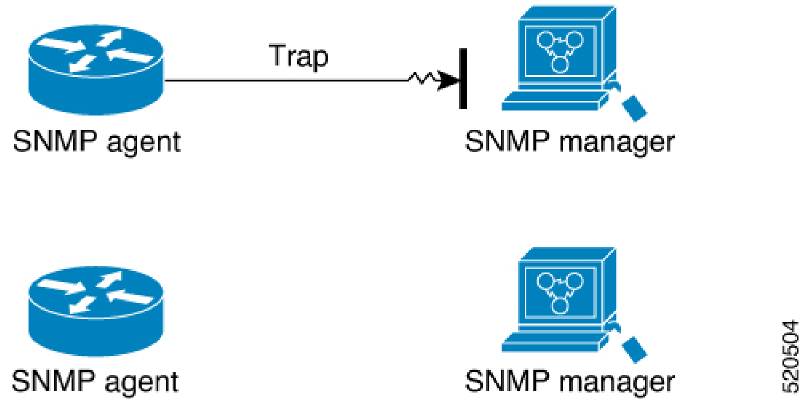


Figure 3: Trap Not Received by the SNMP Manager

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



SNMP Versions

Cisco IOS XR software supports the following versions of SNMP:

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

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

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

SNMPv3 is a security model. A *security model* is an authentication strategy that is set up for a user and the group in which the user resides. A *security level* is the permitted level of security within a security model. A combination of a security model and a security level will determine which security mechanism is employed when an SNMP packet is handled. See Table 4: SNMP Security Models and Levels, on page 10 for a list of security levels available in SNMPv3. The SNMPv3 feature supports RFCs 3411 to 3418.

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

Comparison of SNMPv1, v2c, and v3

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

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

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

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

Table 3: SNMPv1, v2c, and v3 Feature Support

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

Security Models and Levels for SNMPv1, v2, v3

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

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

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

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

Table 4: SNMP Security Models and Levels

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

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

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

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

SNMPv3 Benefits

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

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

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

SNMPv3 Costs

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

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

Table 5: Order of Response Times from Least to Greatest

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

User-Based Security Model

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

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

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

USM uses two authentication protocols:

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

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

View-Based Access Control Model

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

MIB Views

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

Access Policy

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

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

IP Precedence and DSCP Support for SNMP

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

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

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

Session MIB support on subscriber sessions

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

SNMP Notifications

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

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

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

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

Figure 4: 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.

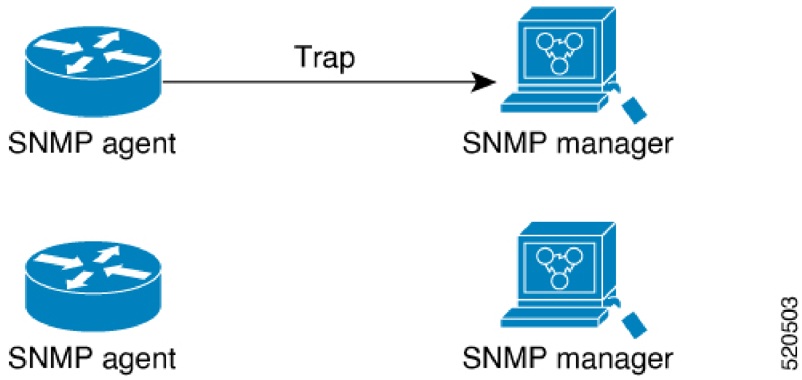
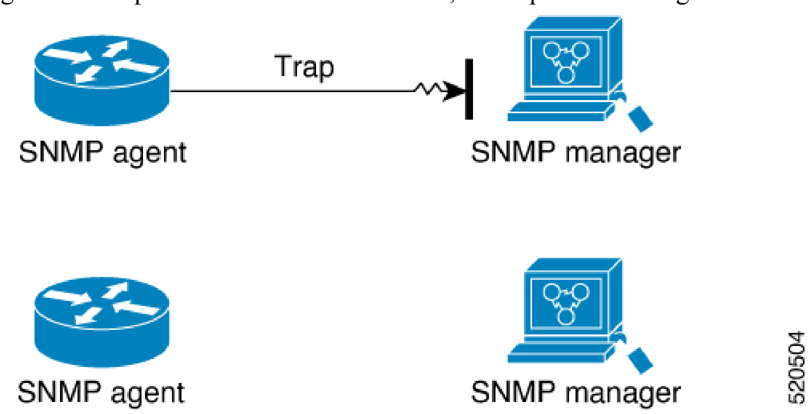


Figure 5: Trap Not Received by the SNMP Manager

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.



Session Types

The supported session types are:

- PPPoE
- IP SUB PKT
- IP SUB DHCP

How to Implement SNMP on Cisco IOS XR Software

This section describes how to implement SNMP.

The **snmp-server** commands enable SNMP on Management Ethernet interfaces by default. For information on how to enable SNMP server support on other inband interfaces, see the *Implementing Management Plane Protection on Cisco IOS XR Software* module in *System Security Configuration Guide for Cisco ASR 9000 Series Routers*.

Configuring SNMPv3

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server view** *view-name oid-tree* {**included** | **excluded**} **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server view view_name 1.3.6.1.2.1.1.5 included`	Creates or modifies a view record.
Step 3	**snmp-server group** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*]	Configures a new SNMP group or a table that maps SNMP users to SNMP views.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(config)# snmp-server group group_name v3 noauth read view_name1 write view_name2`	
Step 4	**snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**priv des56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*] **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server user noauthuser group_name v3`	Configures a new user to an SNMP group. **Note** Only one remote host can be assigned to the same username for SNMP version 3. If you configure the same username with different remote hosts, only the last username and remote host combination will be accepted and will be seen in the **show running** configuration. In the case of multiple SNMP managers, multiple unique usernames are required. **Note** When you execute an SNMP bulk request using the **snmpbulkget** command to an unavailable MIB, it provides a next available MIB.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring SNMP Trap Notifications

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

Note You can omit Step 3, on page 15 if you have already completed the steps documented under the Configuring SNMPv3, on page 15 task.

SUMMARY STEPS

1. **configure**
2. **snmp-server group** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*]

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server group** *name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *view*] [**write** *view*] [**notify** *view*] [*access-list-name*] **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server group group_name v3 noauth read view_name1 write view_name2`	Configures a new SNMP group or a table that maps SNMP users to SNMP views.
Step 3	**snmp-server user** *username groupname* {**v1** | **v2c** | **v3** [**auth** {**md5** | **sha**} {**clear** | **encrypted**} *auth-password* [**priv des56** {**clear** | **encrypted**} *priv-password*]]} [*access-list-name*] **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server user noauthuser group_name v3`	Configures a new user to an SNMP group. **Note** Only one remote host can be assigned to the same username for SNMP version 3. If you configure the same username with different remote hosts, only the last username and remote host combination will be accepted and will be seen in the **show running** configuration. In the case of multiple SNMP managers, multiple unique usernames are required. **Note** When you execute an SNMP bulk request using the **snmpbulkget** command to an unavailable MIB, it provides a next available MIB.
Step 4	**snmp-server host** *address* [**traps**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server host 12.26.25.61 traps version 3 noauth userV3noauth`	Specifies SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the notifications.
Step 5	**snmp-server traps** [*notification-type*] **Example:**	Enables the sending of trap notifications and specifies the type of trap notifications to be sent.

	Command or Action	Purpose
	`RP/0/RP0/CPU0: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.
Step 6	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 7	(Optional) **show snmp host** **Example:** `RP/0/RSP0/CPU0:router# show snmp host`	Displays information about the configured SNMP notification recipient (host), port number, and security model.

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# configure`	

Step 2	(Optional) **snmp-server contact** *system-contact-string*
	Example:
	`RP/0/RSP0/CPU0:router(config)# snmp-server contact`
	`Dial System Operator at beeper # 27345`
		Sets the system contact string.

Step 3	(Optional) **snmp-server location** *system-location*
	Example:
	`RP/0/RSP0/CPU0:router(config)# snmp-server location`
	`Building 3/Room 214`
		Sets the system location string.

Step 4	(Optional) **snmp-server chassis-id** *serial-number* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server chassis-id 1234456`	Sets the system serial number.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Defining the Maximum SNMP Agent Packet Size

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

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

SUMMARY STEPS

1. **configure**
2. (Optional) **snmp-server packetsize** *byte-count*

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	(Optional) **snmp-server packetsize** *byte-count* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server packetsize 1024`	Sets the maximum packet size.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Changing Notification Operation Values

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	(Optional) **snmp-server trap-source** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server trap-source POS 0/0/1/0`	Specifies a source interface for trap notifications.
Step 3	(Optional) **snmp-server queue-length** *length* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server queue-length 20`	Establishes the message queue length for each notification.
Step 4	(Optional) **snmp-server trap-timeout** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server trap-timeout 20`	Defines how often to resend notifications on the retransmission queue.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Setting IP Precedence and DSCP Values

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

Before you begin

SNMP must be configured.

SUMMARY STEPS

1. **configure**

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	Use one of the following commands: • **snmp-server ipv4 precedence** *value* • **snmp-server ipv4 dscp** *value* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server dscp 24`	Configures an IP precedence or IP DSCP value for SNMP traffic.
Step 3	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring MIB Data to be Persistent

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **snmp-server entityindex persist** **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server entityindex persist`	Enables the persistent storage of ENTITY-MIB data.
Step 2	(Optional) **snmp-server mibs cbqosmib persist** **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server mibs cbqosmib persist`	Enables persistent storage of the CISCO-CLASS-BASED-QOS-MIB data.
Step 3	(Optional) **snmp-server cbqosmib cache refresh time** *time* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server mibs cbqosmib cache refresh time 45`	Enables QoS MIB caching with a specified cache refresh time.
Step 4	(Optional) **snmp-server cbqosmib cache service-policy count** *count* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server mibs cbqosmib cache service-policy count 50`	Enables QoS MIB caching with a limited number of service policies to cache.
Step 5	**snmp-server ifindex persist** **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server ifindex persist`	Enables ifIndex persistence globally on all Simple Network Management Protocol (SNMP) interfaces.

Configuring LinkUp and LinkDown Traps for a Subset of Interfaces

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

Before you begin

SNMP must be configured.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**snmp-server interface subset** *subset-number* **regular-expression** *expression*
	Example:
	`RP/0/RSP0/CPU0:router(config)# snmp-server interface subset 10`
	`    regular-expression "^Gig[a-zA-Z]+[0-9/]+\."`
	`RP/0/RSP0/CPU0:router(config-snmp-if-subset)#`
		Enters snmp-server interface mode for the interfaces identified by the regular expression.
		The subset-number argument identifies the set of interfaces, and also assigns a priority to the subset in the event that an interface is included in more than one subset. Lower numbers have higher priority and their configuration takes precedent over interface subsets with higher numbers.
		The *expression* argument must be entered surrounded by double quotes.
		Refer to the *Understanding Regular Expressions, Special Characters, and Patterns* module in *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide* for more information regarding regular expressions.

Step 3	**notification linkupdown disable** **Example:** `RP/0/RSP0/CPU0:router(config-snmp-if-subset)# notification linkupdown disable`	Disables linkUp and linkDown traps for all interfaces being configured. To enable previously disabled interfaces, use the **no** form of this command.

	Command or Action	Purpose
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes, and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration mode, without committing the configuration changes.
Step 5	(Optional) **show snmp interface notification subset** *subset-number* **Example:** `RP/0/RSP0/CPU0:router# show snmp interface notification subset 10`	Displays the linkUp and linkDown notification status for all interfaces identified by the subset priority.
Step 6	(Optional) **show snmp interface notification regular-expression** *expression* **Example:** `RP/0/RSP0/CPU0:router# show snmp interface notification regular-expression "^Gig[a-zA-Z]+[0-9/]+\."`	Displays the linkUp and linkDown notification status for all interfaces identified by the regular expression.
Step 7	(Optional) **show snmp interface notification** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router# show snmp interface notification tengige 0/4/0/3.10`	Displays the linkUp and linkDown notification status for the specified interface.

Configuration Examples for Implementing SNMP

Configuring SNMPv3: Examples

Setting an Engine ID

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

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

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

Verifying the Identification of the Local SNMP Engines

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

```
config
  show snmp engineid

  SNMP engineID 00000009000000a1ffffffff
```

Creating a View

There are two ways to create a view:

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

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

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

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

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

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

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

Verifying Configured Views

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

```
RP/0/RSP0/CPU0:router# show snmp view

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

```
  SNMP_VIEW1 1.3.6.1.2.1.1.5 - excluded nonVolatile active
```

Creating Groups

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

```
RP/0/RSP0/CPU0:router(config)# snmp-server group group-name v3 auth
```

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

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

Verifying Groups

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

```
RP/0/RSP0/CPU0:router# show snmp group

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

Creating and Verifying Users

Given the following SNMPv3 view and SNMPv3 group configuration:

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

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

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

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

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

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

```
snmp-server user username  nervectrgrp remote 10.69.236.146 udp-port 162 v3 auth sha
<password> priv aes 128 <password>
snmp-server user username  nervectrgrp remote 10.214.127.2 udp-port 162 v3 auth sha <password>
 priv aes 128  <password>
snmp-server user username  nervectrgrp remote 10.69.236.147 udp-port 162 v3 auth sha
<password> priv aes 128 <password>
RP/0/RSP0/CPU0:router# show run snmp-server user

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

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

:

```
snmp-server user username1  nervectrgrp remote 10.69.236.146 udp-port 162 v3 auth sha
<password> priv aes 128 <password>
snmp-server user username2  nervectrgrp remote 10.214.127.2 udp-port 162 v3 auth sha
<password> priv aes 128  <password>
snmp-server user username3  nervectrgrp remote 10.69.236.147 udp-port 162 v3 auth sha
<password> priv aes 128 <password>
RP/0/RSP0/CPU0:router# show run snmp-server user

  snmp-server user batmanusr1 nervectrgrp remote 10.69.236.146 udp-port 162 v3 auth sha
encrypted <password> priv aes 128 encrypted <password>
  snmp-server user batmanusr2 nervectrgrp remote 10.214.127.2 udp-port 162 v3 auth sha
encrypted <password> priv aes 128 encrypted <password>
  snmp-server user batmanusr3 nervectrgrp remote 10.69.236.147 udp-port 162 v3 auth sha
encrypted <password> priv aes 128 encrypted <password>
```

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

```
RP/0/RSP0/CPU0:router# show snmp user

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

Given the following SNMPv3 view and SNMPv3 group configuration:

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

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

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

Given the following SNMPv3 view and SNMPv3 group configuration:

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

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

```
RP/0/RSP0/CPU0:router(config)# snmp-server user authuser group_name v3 auth md5 clear
auth_passwd
```

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

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

```
RP/0/RSP0/CPU0:router# show snmp user

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

Given the following SNMPv3 view and SNMPv3 group configuration:

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

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

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

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

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

```
RP/0/RSP0/CPU0:router# show snmp user

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

Configuring Trap Notifications: Example

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

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

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

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

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

```
config
  show snmp host

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

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

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

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

Setting an IP Precedence Value for SNMP Traffic: Example

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

```
configure
  snmp-server ipv4 precedence 7
  exit

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

Setting an IP DSCP Value for SNMP Traffic: Example

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

```
configure
  snmp-server ipv4 dscp 45
  exit

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

SNMP Context Mapping Configuration

Configuration of VRF Aware SNMP Context for Polling BGP Data

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

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

Before you begin

- Ensure that MIB implementation is VRF-aware.

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

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

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

Verification

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

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

Configuration of OSPF processes Using SNMP Context

The following example configures data polling from two OSPF processes.

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

```
 snmp context ctx2
 area 0
  interface GigabitEthernet0/2/0/1
  !
 !
!
```

Configuration of OSPF Neighbour in VRF

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

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

Additional References

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

Related Documents

Related Topic	Document Title
Cisco IOS XR SNMP commands	*SNMP Server Commands on the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*
MIB information	*Cisco ASR 9000 Series Aggregation Services Routers MIB Specifications Guide*
Cisco IOS XR commands	*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*
Getting started with Cisco IOS XR software	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Information about user groups and task IDs	*Configuring AAA Services on the Cisco ASR 9000 Series Router* module of *System Security Configuration Guide for Cisco ASR 9000 Series Routers*
Cisco IOS XR Quality of Service	*Modular QoS Configuration Guide for Cisco ASR 9000 Series Routers*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 4

Configuring Object Tracking

This module describes the configuration of object tracking on your Cisco IOS XR network. For complete descriptions of the commands listed in this module, see Related Documents, on page 51. To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 6: Feature History Table

Feature Name	Release Information	Description
Enhanced Object Tracking for Loopback Interfaces	Release 7.10.1	You can now track an object (for example, an interface) and configure to error-disable a loopback interface when the tracked object state changes. This is beneficial when the loopback interface is the source of multiple tunnels and all tunnels need to be brought down simultaneously. Earlier, you couldn't error-disable loopback interface.
Enhanced Object Tracking	Release 6.4.2	The Enhanced Object Tracking feature is introduced. The ability to error-disable interfaces is added based on the state of objects that are tracked.
Enhanced Object Tracking	Release 4.2.1	The ability to create a tracked list based on a threshold percentage or weight was added.
Enhanced Object Tracking	Release 4.0.0	This feature was introduced.

This module contains the following topics:

Prerequisites for Implementing Object Tracking

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

Information About Object Tracking

Object tracking is a mechanism for tracking an object to take any client action on another object as configured by the client. The object on which the client action is performed may not have any relationship to the tracked objects. The client actions are performed based on changes to the properties of the object being tracked.

You can identify each tracked object by a unique name that is specified by the track command in the configuration mode.

The tracking process periodically polls the tracked object and reports any changes to its state. The state of the tracked objects can be up or down. The polling occurs either immediately or after a delay of a configured period.

You can also track multiple objects by a list. You can use a flexible method for combining objects with Boolean logic. This functionality includes:

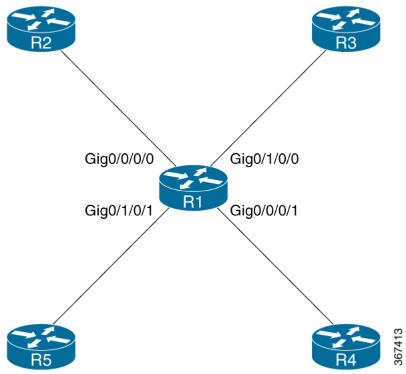
- **Boolean AND function**—When a tracked list has been assigned a Boolean AND function, each object that is defined within a subset must be in an "up" state. This condition enables the tracked object to be in the "up" state.
- **Boolean OR function**—When the tracked list has been assigned a Boolean OR function, at least one object that is defined within a subset must also be in an "up" state. This condition enables the tracked object to be in the "up" state.

Enhanced Object Tracking allows you to extend the track function to implement actions. These actions are triggered when the state of the object that is being tracked changes to "up" or "down". Based on the track state, you can error-disable one or more specified interfaces. Unless you configure the **auto-recover** keyword, the interfaces remain disabled even after the track state changes to the original state. You can configure **auto-recover** for each **action** configuration on a track.

In Figure 1, tracks named track1 and track2 are configured on router R1 to track the line protocol state of interfaces, GigabitEthernet0/0/0/1 and GigabitEthernet0/1/0/1 respectively. A track that is named track3 is configured to track track1 and track2 tracks with the Boolean logic AND. Therefore, track3 goes down if one or both the tracks, track1 and track2, go down. Track3 is also configured with the **action** command to put the interfaces GigabitEthernet0/0/0/0 and GigabitEthernet0/1/0/0 in a disabled state when track3 goes down.

Once the interfaces are error-disabled, they remain in the error-disabled state even if the track state changes to the "up" state. This is the default behaviour. To change this default behaviour, you can optionally configure the **auto-recover** keyword in the **action** command. If you configure the optional **auto-recover** keyword, the error-disabled state on the interfaces is cleared when the track state changes to the "up" state.

Figure 6: Enhanced Object Tracking



The same capability is available to progammatically manage the loopback interface as well. You can error-disable the loopback interface when the tracked-object state goes down.

For example, assume the loopback interface is the source of multiple tunnels. You can configure a track to monitor the line-protocol state of another interface with an action to error-disable the loopback interface when the tracked interface state goes down. When the line-protocol state of the tracked interface goes down, the loopback interface gets error-disabled, thereby bringing down all the associated tunnels. This avoids shutting down each tunnel interface individually.

As with the other interfaces, you can optionally configure the **auto-recover** keyword in the **action** command to clear the error-disabled state of the loopback interface when the track state changes back up. If this keyword is not configured, the loopback interface remains down even when the tracked interface state changes back up.

Restrictions for Enhanced Object Tracking

- You can perform Enhanced Object Tracking on physical interfaces, bundled interfaces, and loopback interface. Other virtual interfaces do not support this.
- The only action you can perform is error-disabling interfaces based on the state of a track (up/down).
- The maximum number of action interfaces that can be added under a single track is 1024.

How to Implement Object Tracking

This section describes the various object tracking procedures.

Tracking the Line Protocol State of an Interface

Perform this task in global configuration mode to track the line protocol state of an interface.

A tracked object is considered up when a line protocol of the interface is up.

After configuring the tracked object, you may associate the interface whose state should be tracked and specify the number of seconds to wait before the tracking object polls the interface for its state.

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type** **line-protocol** **state**
4. **interface** *type* *interface-path-id*
5. **exit**
6. (Optional) **delay** {**up** *seconds* | **down** *seconds*}
7. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**track** *track-name* **Example:** `RP/0/RSP0/CPU0:router(config)# track track1`	Enters track configuration mode. • *track-name*—Specifies a name for the object to be tracked. **Note** Special characters are not allowed in a *track-name*.
Step 3	**type** **line-protocol** **state** **Example:** `RP/0/RSP0/CPU0:router(config-track)# type line-protocol state`	Creates a track based on the line protocol of an interface.
Step 4	**interface** *type* *interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-track-line-prot)# interface atm 0/2/0/0.1`	Specifies the interface to track the protocol state. • *type*—Specifies the interface type. For more information, use the question mark (?) online help function. • *interface-path-id*—Identifies a physical interface or a virtual interface. **Note** Use the **show interfaces** command to see a list of all possible interfaces currently configured on the router. **Note** The loopback and null interfaces are always in the up state and, therefore, cannot be tracked.

	Command or Action	Purpose
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-track-line-prot)# exit`	Exits the track line protocol configuration mode.
Step 6	(Optional) **delay** {**up** *seconds* | **down** *seconds*} **Example:** `RP/0/RSP0/CPU0:router(config-track)# delay up 10`	Schedules the delay that can occur between tracking whether the object is up or down.

Step 7	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-track)# end`
	or
	`RP/0/RSP0/CPU0:router(config-track)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Tracking IP Route Reachability

When a host or a network goes down on a remote site, routing protocols notify the router and the routing table is updated accordingly. The routing process is configured to notify the tracking process when the route state changes due to a routing update.

A tracked object is considered up when a routing table entry exists for the route and the route is accessible.

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type route reachability**
4. Use one of the following commands:
   - **vrf** *vrf-table-name*

- **route ipv4** *IP-prefix/mask*

5. **exit**
6. (Optional) **delay** { **up** *seconds* | **down** *seconds* }
7. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**track** *track-name* **Example:** `RP/0/RSP0/CPU0:router(config)# track track1`	Enters track configuration mode. • *track-name*—Specifies a name for the object to be tracked. **Note** Special characters are not allowed in a *track-name*.
Step 3	**type route reachability** **Example:** `RP/0/RSP0/CPU0:router(config-track)# type route reachability vrf internet`	Configures the routing process to notify the tracking process when the state of the route changes due to a routing update.

Step 4	Use one of the following commands:
	• **vrf** *vrf-table-name*
	• **route ipv4** *IP-prefix/mask*
	
	Example:
	
	`RP/0/RSP0/CPU0:router(config-track-route)# vrf vrf-table-4`
	
	or
	
	`RP/0/RSP0/CPU0:router(config-track-route)# route ipv4 10.56.8.10/16`
		Configures the type of IP route to be tracked, which can consist of either of the following, depending on your router type:
		• *vrf-table-name*—A VRF table name.
		• *IP-prefix/mask*—An IP prefix consisting of the network and subnet mask (for example, 10.56.8.10/16).

Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-track-line-prot)# exit`	Exits the track line protocol configuration mode.
Step 6	(Optional) **delay** { **up** *seconds*	**down** *seconds* } **Example:** `RP/0/RSP0/CPU0:router(config-track)# delay up 10`	Schedules the delay that can occur between tracking whether the object is up or down.

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

Building a Track Based on a List of Objects

Perform this task in the global configuration mode to create a tracked list of objects (which, in this case, are lists of interfaces or prefixes) using a Boolean expression to determine the state of the list.

A tracked list contains one or more objects. The Boolean expression enables two types of calculations by using either AND or OR operators. For example, when tracking two interfaces, using the AND operator, up means that *both* interfaces are up, and down means that *either* interface is down.

Note An object must exist before it can be added to a tracked list.

The NOT operator is specified for one or more objects and negates the state of the object.

After configuring the tracked object, you must associate the interface whose state should be tracked and you may optionally specify the number of seconds to wait before the tracking object polls the interface for its state.

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type list boolean** { **and** | **or** }
4. **object** *object-name* [**not**]
5. **exit**
6. (Optional) **delay** {**up** *seconds*|**down** *seconds*}
7. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**track** *track-name* **Example:** `RP/0/RSP0/CPU0:router(config)# track track1`	Enters track configuration mode. • *track-name*—Specifies a name for the object to be tracked. **Note** Special characters are not allowed in a *track-name*.
Step 3	**type** **list** **boolean** { **and** | **or** } **Example:** `RP/0/RSP0/CPU0:router(config-track-list)# type list boolean and`	Configures a Boolean list object and enters track list configuration mode. • **boolean**—Specifies that the state of the tracked list is based on a Boolean calculation. • **and**—Specifies that the list is up if all objects are up, or down if one or more objects are down. For example when tracking two interfaces, up means that both interfaces are up, and down means that either interface is down. • **or**—Specifies that the list is up if at least one object is up. For example, when tracking two interfaces, up means that either interface is up, and down means that both interfaces are down.
Step 4	**object** *object-name* [**not**] **Example:** `RP/0/RSP0/CPU0:router(config-track-list)# object 3 not`	Specifies the object to be tracked by the list • *obect-name*—Name of the object to track. • **not**—Negates the state of the object.
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-track-line-prot)# exit`	Exits the track line protocol configuration mode.
Step 6	(Optional) **delay** {**up** *seconds* | **down** *seconds*} **Example:** `RP/0/RSP0/CPU0:router(config-track)# delay up 10`	Schedules the delay that can occur between tracking whether the object is up or down.
Step 7	Use one of the following commands: • **end** • **commit**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	Example:
	`RP/0/RSP0/CPU0:router(config-track)# end`
	or
	`RP/0/RSP0/CPU0:router(config-track)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Building a Track Based on a List of Objects - Threshold Percentage

Perform this task in the global configuration mode to create a tracked list of objects (which, in this case, are lists of interfaces or prefixes) using a threshold percentage to determine the state of the list.

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type list threshold percentage**
4. **object** *object-name*
5. **threshold percentage up** *percentage* **down** *percentage*
6. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**track** *track-name* **Example:**	Enters track configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# track track1`	• *track-name*—Specifies a name for the object to be tracked. **Note** Special characters are not allowed in a *track-name*.
Step 3	**type list threshold percentage** **Example:** `RP/0/RSP0/CPU0:router(config-track-list)# type list threshold percentage`	Configures a track of type threshold percentage list.

Step 4	**object** *object-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-track-list-threshold)# object 1`
	`RP/0/RSP0/CPU0:router(config-track-list-threshold)# object 2`
	`RP/0/RSP0/CPU0:router(config-track-list-threshold)# object 3`
	`RP/0/RSP0/CPU0:router(config-track-list-threshold)# object 4`
		Configures object 1, object 2, object 3 and object 4 as members of track type track1.

Step 5	**threshold percentage up** *percentage* **down** *percentage*
	Example:
	`RP/0/RSP0/CPU0:router(config-track-list-threshold)# threshold`
	`percentage up 50 down 33`
		Configures the percentage of objects that need to be UP or DOWN for the list to be considered UP or Down respectively.
		For example, if object 1, object 2, and object 3 are in the UP state and object 4 is in the DOWN state, the list is considered to be in the UP state.

Step 6	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-track)# end`
	or
	`RP/0/RSP0/CPU0:router(config-track)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Building a Track Based on a List of Objects - Threshold Weight

Perform this task in the global configuration mode to create a tracked list of objects (which, in this case, are lists of interfaces or prefixes) using a threshold weight to determine the state of the list.

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type list threshold weight**
4. **object** *object-name* **weight** *weight*
5. **threshold weight up** *weight* **down** *weight*
6. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**track** *track-name* **Example:** `RP/0/RSP0/CPU0:router(config)# track track1`	Enters track configuration mode. • *track-name*—Specifies a name for the object to be tracked. **Note** Special characters are not allowed in a *track-name*.
Step 3	**type list threshold weight** **Example:** `RP/0/RSP0/CPU0:router(config-track-list)# type list threshold weight`	Configures a a track of type, threshold weighted list.
Step 4	**object** *object-name* **weight** *weight* **Example:**	Configures object 1, object 2 and object 3 as members of track t1 and with weights 10, 5 and 3 respectively.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-track-list-threshold)# object 1 weight 10`
	`RP/0/RSP0/CPU0:router(config-track-list-threshold)# object 2 weight 5`
	`RP/0/RSP0/CPU0:router(config-track-list-threshold)# object 3 weight 3`	

Step 5	**threshold weight up** *weight* **down** *weight* **Example:** `RP/0/RSP0/CPU0:router(config-track-list-threshold)# threshold weight up 10 down 5`	Configures the range of weights for the objects that need to be UP or DOWN for the list to be considered UP or DOWN respectively. In this example, the list is considered to be in the DOWN state because objects 1 and 2 are in the UP state and the cumulative weight is 15 (not in the 10-5 range).

Step 6	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-track)# end`
	or
	`RP/0/RSP0/CPU0:router(config-track)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Tracking IPSLA Reachability

Use this task to enable the tracking of the return code of IP service level agreement (SLA) operations.

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type rtr** *ipsla-no* **reachability**
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**track** *track-name* **Example:** `RP/0/RSP0/CPU0:router(config)# track t1`	Enters track configuration mode. **Note** Special characters are not allowed in a *track-name*.
Step 3	**type rtr** *ipsla-no* **reachability** **Example:** `RP/0/RSP0/CPU0:router(config-track)# type rtr 100 reachability`	Specifies the IP SLA operation ID to be tracked for reachability. Values for the *ipsla-no* can range from1 to 2048.
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring IPSLA Tracking: Example

This example shows the configuration of IPSLA tracking:

```
RP/0/RSP0/CPU0:router(config)# track track1
RP/0/RSP0/CPU0:router(config-track)# type rtr 1 reachability
RP/0/RSP0/CPU0:router(config-track)# delay up 5
RP/0/RSP0/CPU0:router(config-track)# delay down 10
```

Configure Enhanced Object Tracking

You can configure tracks with the **action** command to enable Enhanced Object Tracking. As a prerequisite, configure the track type that is to be tracked.

The following example shows how to configure the **action** command on a track based on the change in state of the track:

```
/* Configure track1 to track line-protocol state of the interface FourHundredGigE0/0/0/1
```

```
*/
Router#configure
Router(config)#track track1
Router(config-track)#type line-protocol state
Router(config-track-line-prot)#interface FourHundredGigE0/0/0/1
Router(config-track-line-prot)#exit
Router(config-track)#exit

/* Configure track2 to track line-protocol state of the interface FourHundredGigE0/1/0/1
*/
Router(config)#track track2
Router(config-track)#type line-protocol state
Router(config-track-line-prot)#interface FourHundredGigE0/1/0/1
Router(config-track-line-prot)#exit
Router(config-track)#exit

/* Configure track3 with boolean AND of track1 state and track2 state. Specify actions to
take when track3 state changes. */
Router(config)#track track3
Router(config-track)#type list boolean and
Router(config-track-list-boolean)#object track1
Router(config-track-list-boolean)#object track2
Router(config-track-line-boolean)#exit
Router(config-track)#action
Router(config-track-action)#track-down error-disable interface FourHundredGigE0/0/0/0
auto-recover
Router(config-track-action)#track-down error-disable interface FourHundredGigE0/1/0/0

/* Configure track4 to track line-protocol state of the HundredGigE0/0/0/35 interface and
take action on loopback interface. */
Router(config)#track track4
Router(config-track)#type line-protocol state
Router(config-track-line-prot)#interface HundredGigE0/0/0/35
Router(config-track-line-prot)#exit

/* Specify action to take when track4 state changes. In this example, the action happens
when the track state changes to up. */
Router(config-track)#action
Router(config-track-action)#track-up error-disable interface Loopback100 auto-recover
Router(config-track)#exit
Router(config)#end
```

The following running configuration example shows you how to configure the **action** command for the scenario described in Figure 1.

```
track track1
 type line-protocol state
  interface FourHundredGigE0/0/0/1
 !
!
track track2
 type line-protocol state
  interface FourHundredGigE0/1/0/1
 !
!
track track3
 type list boolean and
  object track1
  object track2
 !
 action
  track-down error-disable interface FourHundredGigE0/0/0/0 auto-recover
  track-down error-disable interface FourHundredGigE0/1/0/0
```

The following example shows you how to configure the **action** command for the loopback interface.

```
track track4
 type line-protocol state
  interface HundredGigE0/0/0/35
 !
 action
  track-up error-disable interface Loopback100 auto-recover
 !
!
```

Verification

To view the state of the track, use the **show track** command.

Initially, let us assume the line-protocol state of FourHundredGigE0/0/0/1 (track1 interface) and FourHundredGigE0/1/0/1 (track2 interface) are up and HundredGigE0/0/0/35 (track4 interface) is down.

```
Router#show track
Track track3
        List boolean and is UP
        7 changes, last change 16:04:28 IST Mon Jul 02 2018
                object track2 UP
                object track1 UP
Track track1
        Interface FourHundredGigE0/0/0/1 line-protocol
        Line protocol is UP
        7 changes, last change 16:04:28 IST Mon Jul 02 2018
Track track2
        Interface FourHundredGigE0/1/0/1 line-protocol
        Line protocol is UP
        7 changes, last change 16:02:41 IST Mon Jul 02 2018

Track track4
        Interface HundredGigE0/0/0/35 line-protocol
        Line protocol is DOWN
        2 changes, last change 06:28:06 UTC Tue Jun 27 2023
        Delay up 0 secs(default), down 0 secs(default)
```

To verify if the interface configured for tracking is error-disabled, use the **show error-disable** command. As none of the track states match the track-action state, there are no error-disabled interfaces.

```
Router#show error-disable
Interface           Error-Disable reason          Retry (s)  Time disabled
-------------------  ------------------------------  ---------  ---------------
There are no interfaces error-disabled matching the given criteria
```

To view the status of all the interfaces of the tracked object, use the **show interface brief** command.

```
Router#show interface brief
Intf Name                    Intf State LineP State    Encap Type   MTU (byte) BW (Kbps)
Lo100                        up         up               Loopback   1500       0
FourHundredGigE0/0/0/0       up         up               ARPA       1514       100000000
FourHundredGigE0/0/0/1       up         up               ARPA       1514       100000000
FourHundredGigE0/1/0/0       up         up               ARPA       1514       100000000
FourHundredGigE0/1/0/1       up         up               ARPA       1514       100000000
HundredGigE0/0/0/35          admin-down admin-down       ARPA       1514       100000000
```

When a track state changes, the corresponding track action happens and the status of the interfaces configured in the action changes. The state of track3 becomes "down" when either track1 state or track2 state becomes "down". The state of track4 becomes "up" when the HundredGigE0/0/0/35 interface comes up. The **show error-disable** command displays the following output when track3 state is down and track4 state is up.

```
Router#show error-disable
Interface           Error-Disable reason            Retry (s)  Time disabled
-------------------  ------------------------------  ---------  ---------------
Loopback100         ot-track-state-change             ---      08:44:07
FH0/0/0/0           ot-track-state-change             ---      08:42:08
FH0/1/0/0           ot-track-state-change             ---      08:42:01
```

When track3 state is down and track4 state is up, the **show interface brief** command displays the following output.

```
Router#show interface brief
Intf Name                 Intf State      LineP State    Encap Type   MTU (byte)   BW (Kbps)
Lo100                      err-disable     admin-down     Loopback    1500          0
FourHundredGigE0/0/0/0    err-disable     admin-down     ARPA        1514          100000000
FourHundredGigE0/0/0/1    err-disable     admin-down     ARPA        1514          100000000
FourHundredGigE0/1/0/0    err-disable     admin-down     ARPA        1514          100000000
FourHundredGigE0/1/0/1    up              up             ARPA        1514          100000000
HundredGigE0/0/0/35       up              up             ARPA        1514          100000000
```

When track3 state comes back up, the error-disable status on the interface FourHundredGigE0/0/0/0 clears. This is because of the **auto-recover** configuration for FourHundredGigE0/0/0/0. However, interface FourHundredGigE0/1/0/0 remains in the error-disable status because **auto-recover** isn't configured on this interface.

Similarly, when track4 state goes down, the error-disable status on the interface Loopback100 clears because of the **auto-recover** configuration for track4.

The change reflects in the output of the **show interface brief** command.

```
RP/0/0/CPU0:ios#show interface brief
Intf Name                 Intf State  LineP State  Encap Type  MTU (byte) BW (Kbps)
Lo100                     up          up           Loopback    1500       0
FourHundredGigE0/0/0/0    up          up           ARPA        1514       100000000
FourHundredGigE0/0/0/1    up          up           ARPA        1514       100000000
FourHundredGigE0/1/0/0    err-disable admin-down   ARPA        1514       100000000
FourHundredGigE0/1/0/1    up          up           ARPA        1514       100000000
HundredGigE0/0/0/35       admin-down  admin-down   ARPA        1514       100000000
```

Configuration Examples for Configuring Object Tracking

Configuring IPSLA Tracking: Example

This example shows the configuration of IPSLA tracking, including the ACL and IPSLA configuration:

ACL configuration:

```
RP/0/RSP0/CPU0:router(config)# ipv4 access-list abf-track
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit any any nexthop track track1 1.2.3.4
```

Object tracking configuration:

```
RP/0/RSP0/CPU0:router(config)# track track1
RP/0/RSP0/CPU0:router(config-track)# type rtr 1 reachability
RP/0/RSP0/CPU0:router(config-track)# delay up 5
RP/0/RSP0/CPU0:router(config-track)# delay down 10
```

IPSLA configuration:

```
RP/0/RSP0/CPU0:router(config)# ipsla
RP/0/RSP0/CPU0:router(config-ipsla)# operation 1
RP/0/RSP0/CPU0:router(config-ipsla-op)# type icmp echo
RP/0/RSP0/CPU0:router(config-ipsla-icmp-echo)# source address 2.3.4.5
RP/0/RSP0/CPU0:router(config-ipsla-icmp-echo)# destination address 1.2.3.4
RP/0/RSP0/CPU0:router(config-ipsla-icmp-echo)# frequency 60
RP/0/RSP0/CPU0:router(config-ipsla-icmp-echo)# exit
RP/0/RSP0/CPU0:router(config-ipsla-op)# exit
RP/0/RSP0/CPU0:router(config-ipsla)# schedule operation 1
RP/0/RSP0/CPU0:router(config-ipsla-sched)# start-time now
RP/0/RSP0/CPU0:router(config-ipsla-sched)# life forever
```

Additional References

The following sections provide references related to implementing object tracking for IPSec network security.

Related Documents

Related Topic	Document Title
IP SLA configuration information	*Implementing IP Service Level Agreements on the Cisco ASR 9000 Series Router* module in *System Monitoring Configuration Guide for Cisco ASR 9000 Series Routers*
IP SLA commands	*IP Service Level Agreement Commands on the Cisco ASR 9000 Series Router* module in *System Monitoring Command Reference for Cisco ASR 9000 Series Routers*
Object tracking commands	*Object Tracking Commands on the Cisco ASR 9000 Series Router* module in *System Management Command Reference for Cisco ASR 9000 Series Routers*

Standards

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

MIBs

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

RFCs

RFCs	Title
RFC 2401	*Security Architecture for the Internet Protocol*

Technical Assistance

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

CHAPTER 5

Configuring Periodic MIB Data Collection and Transfer

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

Table 7: Feature History for Periodic MIB Data Collection and Transfer

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

This module contains the following topics:

Prerequisites for Periodic MIB Data Collection and Transfer

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

Information About Periodic MIB Data Collection and Transfer

SNMP Objects and Instances

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

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

Bulk Statistics Object Lists

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

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

Bulk Statistics Schemas

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

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

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

Bulk Statistics Transfer Options

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

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

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

Benefits of Periodic MIB Data Collection and Transfer

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

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

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

How to Configure Periodic MIB Data Collection and Transfer

Configuring a Bulk Statistics Object List

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server mib bulkstat object-list** *list-name* **Example:** `snmp-server mib bulkstat object-list ifMib`	Defines an SNMP bulk statistics object list and enters bulk statistics object list configuration mode.
Step 3	**add** {**oid** | *object-name*} **Example:** `RP/0/RSP0/CPU0:router(config-bulk-objects)# add`	Adds a MIB object to the bulk statistics object list. Repeat as desired until all objects to be monitored in this list are added.

	Command or Action	Purpose

	`1.3.6.1.2.1.2.2.1.11`
	`RP/0/RSP0/CPU0:router(config-bulk-objects)# add ifAdminStatus`
	`RP/0/RSP0/CPU0:router(config-bulk-objects)# add ifDescr`
		Note All the objects in a bulk statistics object list have to be indexed by the same MIB index. However, the objects in the object list do not need to belong to the same MIB or MIB table.
		
		When specifying an object name instead of an OID (using the add command), only object names with mappings shown in the **show snmp mib object** command output can be used.

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

What to do next

Configure a bulk statistics schema.

Configuring a Bulk Statistics Schema

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

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**snmp-server mib bulkstat schema** *schema-name*
	Example:
	`RP/0/RSP0/CPU0:router(config)# snmp-server mib bulkstat schema intE0`
	`RP/0/RSP0/CPU0:router(config-bulk-sc)#`
		Names the bulk statistics schema and enters bulk statistics schema mode.

Step 3	**object-list** *list-name* **Example:** `RP/0/RSP0/CPU0:router(config-bulk-sc)# object-list ifMib`	Specifies the bulk statistics object list to be included in this schema. Specify only one object list per schema. If multiple object-list commands are executed, the earlier ones are overwritten by newer commands.

Step 4	Do one of the following:
	• **instance exact** {**interface** *interface-id* [**sub-if**] | **oid** *oid*}
	• **instance wild** {**interface** *interface-id* [**sub-if**] | **oid** *oid*}
	• **instance range start** *oid* **end** *oid*
	• **instance repetition** *oid* **max** *repeat-number*
	Example:
	`RP/0/RSP0/CPU0:router(config-bulk-sc)# instance wild oid 1`
	or
	`RP/0/RSP0/CPU0:router(config-bulk-sc)# instance exact interface FastEthernet 0/1.25`
	or
	`RP/0/RSP0/CPU0:router(config-bulk-sc)# instance range start 1 end 2`
	or
	`RP/0/RSP0/CPU0:router(config-bulk-sc)# instance repetition 1 max 4`
		Specifies the instance information for objects in this schema:
		• The **instance exact** command indicates that the specified instance, when appended to the object list, represents the complete OID.
		• The **instance wild** command indicates that all subindices of the specified OID belong to this schema. The wild keyword allows you to specify a partial, “wild carded” instance.
		• The **instance range** command indicates a range of instances on which to collect data.
		• The **instance repetition** command indicates data collection to repeat for a certain number of instances of a MIB object.
		Note Only one **instance** command can be configured per schema. If multiple **instance** commands are executed, the earlier ones are overwritten by new commands.

Step 5	**poll-interval** *minutes* **Example:** `RP/0/RSP0/CPU0:router(config-bulk-sc)# poll-interval 10`	Sets how often data should be collected from the object instances specified in this schema, in minutes. The default is once every 5 minutes. The valid range is from 1 to 20000.

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

What to do next

Configure the bulk statistics transfer options.

Configuring Bulk Statistics Transfer Options

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

Before you begin

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**snmp-server mib bulkstat transfer-id** *transfer-id* **Example:** `RP/0/RSP0/CPU0:router(config)# snmp-server mib bulkstat transfer bulkstat1`	Identifies the transfer configuration with a name (*transfer-id* argument) and enters bulk statistics transfer configuration mode.
Step 3	**buffer-size** *bytes* **Example:** `RP/0/RSP0/CPU0:router(config-bulk-tr)# buffersize 3072`	(Optional) Specifies the maximum size for the bulk statistics data file, in bytes. The valid range is from 1024 to 2147483647 bytes. The default buffer size is 2048 bytes. **Note** If the maximum buffer size for a bulk statistics file is reached before the transfer interval time expires, all additional data received is deleted. To correct this behavior, you can decrease the polling frequency, or increase the size of the bulk statistics buffer.
Step 4	**format** {**bulkBinary** | **bulkASCII** | **schemaASCII**} **Example:** `RP/0/RSP0/CPU0:router(config-bulk-tr)# format schemaASCII`	(Optional) Specifies the format of the bulk statistics data file (VFile). The default is schemaASCII. **Note** Transfers can only be performed using schemaASCII (cdcSchemaASCII) format. SchemaASCII is a human-readable format that contains parser-friendly hints for parsing data values.

Step 5	**schema** *schema-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-bulk-tr)# schema ATM2/0-IFMIB`
	`RP/0/RSP0/CPU0:router(config-bulk-tr)# schema ATM2/0-CAR`
	`RP/0/RSP0/CPU0:router(config-bulk-tr)# schema Ethernet2/1-IFMIB`
		Specifies the bulk statistics schema to be transferred. Repeat this command as desired. Multiple schemas can be associated with a single transfer configuration; all collected data are placed in a single bulk data file (VFile).

Step 6	**transfer-interval** *minutes* **Example:** RP/0/RSP0/CPU0:router `RP/0/RSP0/CPU0:router(config-bulk-tr)# transfer-interval 20`	(Optional) Specifies how often the bulk statistics file are transferred, in minutes. The default value is once every 30 minutes. The transfer interval is the same as the collection interval.

	Command or Action	Purpose

Step 7	**url primary** *url*
	Example:
	`RP/0/RSP0/CPU0:router(config-bulk-tr)# url primary`
	`ftp://user:password@host/folder/bulkstat1`
		Specifies the network management system (host) that the bulk statistics data file is transferred to, and the protocol to use for transfer. The destination is specified as a Uniform Resource Locator (URL). FTP or TFTP can be used for the bulk statistics file transfer.

Step 8	**url secondary** *url* **Example:** `RP/0/RSP0/CPU0:router(config-bulk-tr)# url secondary tftp://10.1.0.1/tftpboot/user/bulkstat1`	(Optional) Specifies a backup transfer destination and protocol for use in the event that transfer to the primary location fails. FTP or TFTP can be used for the bulk statistics file transfer.
Step 9	**retry** *number* **Example:** `RP/0/RSP0/CPU0:router(config-bulk-tr)# retry 1`	(Optional) Specifies the number of transmission retries. The default value is 0 (in other words, no retries). If an attempt to send the bulk statistics file fails, the system can be configured to attempt to send the file again using this command. One retry includes an attempt first to the primary destination then, if the transmission fails, to the secondary location. For example, if the retry value is 1, an attempt is made first to the primary URL, then to the secondary URL, then to the primary URL again, then to the secondary URL again. The valid range is from 0 to 100. If all retries fail, the next normal transfer occurs after the configured transfer-interval time.
Step 10	**retain** *minutes* **Example:** `RP/0/RSP0/CPU0:router(config-bulk-tr)# retain 60`	(Optional) Specifies how long the bulk statistics file should be kept in system memory, in minutes, after the completion of the collection interval and a transmission attempt is made. The default value is 0. Zero (0) indicates that the file is deleted immediately after the transfer is attempted. The valid range is from 0 to 20000. **Note** If the retry command is used, you should configure a retain interval larger than 0. The interval between retries is the retain interval divided by the retry number. For example, if **retain 10** and **retry 2** are configured, two retries are attempted once every 5 minutes. Therefore, if retain 0 is configured, no retries are attempted.
Step 11	**enable** **Example:** `RP/0/RSP0/CPU0:router(config-bulk-tr)# enable`	Begins the bulk statistics data collection and transfer process for this configuration. • For successful execution of this action, at least one schema with non-zero number of objects must be configured. • Periodic collection and file transfer begins only if this command is configured. Conversely, the **no enable**

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

What to do next

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

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

Monitoring Periodic MIB Data Collection and Transfer

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show snmp mib bulkstat transfer** *transfer-name*	(Optional) The show command for this feature lists all bulk statistics virtual files (VFiles) on the system that have

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

show snmp mib bulkstat transfer Sample Output

```
RP/0/RSP0/CPU0:router# show snmp mib bulkstat transfer

Transfer Name : ifmib
 Retained files

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

Periodic MIB Data Collection and Transfer: Example

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

```
snmp-server mib bulkstat object-list cempo
add cempMemPoolName
add cempMemPoolType
!
snmp-server mib bulkstat schema cempWild
object-list cempo
```

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

This example shows sample bulk statistics file content:

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

CHAPTER 6

Configuring Flexible Command Line Interface

This module describes how to configure and use flexible command line interface (CLI) configuration groups.

Table 8: Feature History for Configuring Flexible CLI Configuration Groups

Release	Modification
Release 4.3.1	Flexible CLI configuration groups were introduced.

This module contains these topics:

Information About Flexible CLI Configuration Groups

Flexible command line interface (CLI) configuration groups provide the ability to minimize repetitive configurations by defining a series of configuration statements in a configuration group, and then applying this group to multiple hierarchical levels in the router configuration tree.

Flexible CLI configuration groups utilize regular expressions that are checked for a match at multiple submodes of the configuration tree based on where the group is applied within the hierarchy. If a match is found at a configuration submode, the corresponding configuration defined in the group is inherited within the matched submode.

Flexible CLI configuration groups also provide an auto-inheritance feature. Auto-inheritance means that any change done to a CLI configuration group is automatically applied to the configuration in any matched submodes that have an apply-group at that hierarchical level. This allows you to make a configuration change or addition once, and have it applied automatically in multiple locations, depending on where you have applied the flexible CLI configuration group.

Flexible Configuration Restrictions

Note these restrictions while using flexible configuration groups:

- Flexible CLI configuration groups are not supported in administration configurations and corresponding apply-groups are not supported in administration configurations.
- Use of preconfigured interfaces in configuration groups is not supported.
- Downgrading from an image that supports configuration groups to an image that does not support them is not supported.
- Access lists, quality of service and route policy configurations do not support the use of configuration groups. Configurations such as these are not valid:

```
group g-not-supported
 ipv4 access-list ...
 !
 ipv6 access-list ...
 !
 ethernet-service access-list ...
 !
 class-map ...
 !
 policy-map ...
 !
 route-policy ...
 !
end-group
```

  You can, however, reference such configurations, as shown in this example:

```
group g-reference-ok
 router bgp 6500
  neighbor 7::7
   remote-as 65000
   bfd fast-detect
   update-source Loopback300
   graceful-restart disable
   address-family ipv6 unicast
    route-policy test1 in
    route-policy test2 out
    soft-reconfiguration inbound always
   !
  !
 !
 interface Bundle-Ether1005
   bandwidth 10000000
   mtu 9188
   service-policy output input_1
   load-interval 30
 !
end-group
```

- Some regular expressions are not supported within groups. For example, ‘?’, ‘|’ and ‘$,’ are not supported within groups. Also some characters such as /d and /w are not supported.

- The choice operator “|” to express multiple match expressions within a regular expression is not supported. For example, these expressions are not supported:

  `Gig.*|Gig.*\..*`—To match on either Gigabit Ethernet main interfaces or Gigabit Ethernet sub-interfaces.

  `Gig.*0/0/0/[1-5]|Gig.*0/0/0/[10-20]`—To match on either Gig.*0/0/0/[1-5] or Gig.*0/0/0/[10-20].

  `'TenGigE.*|POS.*`—To match on either TenGigE.* or POS.* .

- Commands that require a node identifier for the **location** keyword are not supported. For example, this configuration is not supported:

```
lpts pifib hardware police location 0/0/CPU0
```

- Overlapping regular expressions within a configuration group for the same configuration are not supported. For example:

```
group G-INTERFACE
interface 'gig.*a.*'
  mtu 1500
!
interface 'gig.*e.* '
  mtu 2000
!
end-group

interface gigabitethernet0/4/1/0
  apply-group G-INTERFACE
```

  This configuration is not permitted because it cannot be determined whether the `interface gigabitethernet0/4/1/0` configuration inherits `mtu 1500` or `mtu 2000`. Both expressions in the configuration group match gigabitethernet0/4/1/0.

- Up to eight configuration groups are permitted on one apply-group command.
- Use multi-line configuration style to configure Flexible CLI configuration groups (like group or apply-group commands) by entering each configuration mode in a separate line, one configuration per line. This is important so that the configuration properties are fully inherited and for better readability during troubleshooting.

  Example for a correct configuration style is:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# router isis IGP
RP/0/RSP0/CPU0:router(config-isis)# interface Ten 0/4/0/0
RP/0/RSP0/CPU0:router(config-isis-if) # address-family ipv4 unicast
RP/0/RSP0/CPU0:router (config-isis-if-af) # metric 123
```

Configuring a Configuration Group

A configuration group includes a series of configuration statements that can be used in multiple hierarchical levels in the router configuration tree. By using regular expressions in a configuration group, you can create generic commands that can be applied in multiple instances.

Use this task to create and use a configuration group.

> **Note** Flexible CLI configurations are not available through the XML interface.

SUMMARY STEPS

1. **configure**
2. **group** *group-name*
3. Enter configuration commands, starting from global configuration mode. Use regular expressions for interface names and other variable instances.
4. **end-group**
5. **apply-group**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**group** *group-name* **Example:** `RP/0/RSP0/CPU0:router(config)# group g-interf`	Specifies a name for a configuration group and enters group configuration mode to define the group.The *group-name* argument can have up to 32 characters and cannot contain any special characters. For information regarding special characters, refer to the *Understanding Regular Expressions, Special Characters, and Patterns* module in the *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*.

Step 3	Enter configuration commands, starting from global configuration mode. Use regular expressions for interface names and other variable instances.
	Example:
	`RP/0/RSP0/CPU0:router(config)# group g-interf`
	`RP/0/RSP0/CPU0:router(config-GRP)# interface 'GigabitEthernet.*'`
	`RP/0/RSP0/CPU0:router(config-GRP-if)# mtu 1500`
		Specifies the configuration statements that you want included in this configuration group.
		For more information regarding the use of regular expressions, see Regular Expressions in Configuration Groups, on page 72. This example is applicable to all Gigabit Ethernet interfaces.

Step 4	**end-group** **Example:** `RP/0/RSP0/CPU0:router(config-GRP-if)# end-group`	Completes the configuration of a configuration group and exits to global configuration mode.
Step 5	**apply-group** **Example:**	Adds the configuration of the configuration group into the router configuration applicable at the location that the group

	Command or Action	Purpose
	RP/0/RSP0/CPU0:router(config)# **interface GigabitEthernet0/2/0/0** RP/0/RSP0/CPU0:router(config-if)# **apply-group g-interf**	is applied. Groups can be applied in multiple locations, and their effect depends on the location and context. The MTU value from the group g-interf is applied to the interface GigabitEthernet0/2/0/0. If this group is applied in global configuration mode, the MTU value is inherited by all Gigabit Ethernet interfaces that do not have an MTU value configured.

Simple Configuration Group: Example

This example shows how to use configuration groups to add a global configuration to the system:

```
RP/0/RSP0/CPU0:router(config)# group g-logging
RP/0/RSP0/CPU0:router(config-GRP)# logging trap notifications
RP/0/RSP0/CPU0:router(config-GRP)# logging console debugging
RP/0/RSP0/CPU0:router(config-GRP)# logging monitor debugging
RP/0/RSP0/CPU0:router(config-GRP)# logging buffered 10000000
RP/0/RSP0/CPU0:router(config-GRP)# end-group

RP/0/RSP0/CPU0:router(config)# apply-group g-logging
```

When this configuration is committed, all commands contained in the g-logging configuration group are committed.

Configuration Group Applied to Different Places: Example

Configuration groups can be applied to different places, and their effect depends on the context within which they are applied. Consider this configuration group:

```
RP/0/RSP0/CPU0:router(config)# group g-interfaces
RP/0/RSP0/CPU0:router(config-GRP)# interface 'FastEthernet.*'
RP/0/RSP0/CPU0:router(config-GRP-if)# mtu 1500
RP/0/RSP0/CPU0:router(config-GRP-if)# exit
RP/0/RSP0/CPU0:router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RSP0/CPU0:router(config-GRP-if)# mtu 1000
RP/0/RSP0/CPU0:router(config-GRP-if)# exit
RP/0/RSP0/CPU0:router(config-GRP)# interface 'POS.*'
RP/0/RSP0/CPU0:router(config-GRP-if)# mtu 2000
RP/0/RSP0/CPU0:router(config-GRP-if)# end-group
```

This group can be applied to Fast Ethernet, Gigabit Ethernet or POS interfaces, and in each instance the applicable MTU is applied. For instance, in this example, the Gigabit Ethernet interface is configured to have an MTU of 1000:

```
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/2/0/0
RP/0/RSP0/CPU0:router(config-if)# apply-group g-interfaces
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 2.2.2.2 255.255.255.0
```

In this example, the Fast Ethernet interface is configured to have an MTU of 1500:

```
RP/0/RSP0/CPU0:router(config)# interface FastEthernet0/2/0/0
RP/0/RSP0/CPU0:router(config-if)# apply-group g-interfaces
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 3.3.3.3 255.255.255.0
```

The same configuration group is used in both cases, but only the applicable configuration statements are used.

Verifying the Configuration of Configuration Groups

Use this task to verify the router configuration using configuration groups:

SUMMARY STEPS

1. **show running-config group** [*group-name*]
2. **show running-config**
3. **show running-config inheritance**
4. **show running-config interface x/y/z inheritance** *config-command*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show running-config group** [*group-name*]
	Example:
	`RP/0/RSP0/CPU0:router# show running-config group`
	
	`group g-int-ge`
	` interface 'GigabitEthernet.*'`
	`  mtu 1000`
	`  negotiation auto`
	` !`
	`end-group`
		Displays the contents of a specific or all configured configuration groups.

Step 2	**show running-config**
	Example:
	`RP/0/RSP0/CPU0:router# show running-config`
	
	`group G-INTERFACE-MTU`
	` interface ‘POS.*’`
	`  mtu 1500`
	` !`
	`end-group`
	
	`interface POS0/4/1/0`
	` apply-group G-INTERFACE-MTU`
	`!`
	`interface POS0/4/1/1`
	` apply-group G-INTERFACE-MTU`
		Displays the running configuration. Any applied groups are displayed. There is no indication as to whether these configuration groups affect the actual configuration or not. In this example, although the group G-INTERFACE-MTU is applied to POS0/4/1/1, the configured MTU value is 2000 and not 1500. This happens if the command **mtu 2000** is configured directly on the interface. An actual configuration overrides a configuration group configuration if they are the same.

	Command or Action	Purpose
	` mtu 2000` `!`	

Step 3	**show running-config inheritance**
	
	Example:
	
	`RP/0/RSP0/CPU0:router# show running-config inheritance`
	`.`
	`.`
	`group G-INTERFACE-MTU`
	` interface 'POS.*'`
	`  mtu 1500`
	` !`
	`end-group`
	`.`
	`.`
	`interface POS0/4/1/0`
	` ## Inherited from group G-INTERFACE-MTU`
	` mtu 1500`
	`!`
	`interface POS0/4/1/1`
	` mtu 2000`
	`!`
	`.`
	`.`
		Displays the inherited configuration where ever a configuration group has been applied.

Step 4	**show running-config interface x/y/z inheritance** *config-command*
	
	Example:
	
	`RP/0/RSP0/CPU0:router# show running-config interface pos0/4/1/0 inheritance [detail]`
	
	`interface POS0/4/1/0`
	` ## Inherited from group G-INTERFACE-MTU`
	` mtu 1500`
		Displays the inherited configuration for a specific configuration command.

Apply Groups Priority Inheritance

The inheritance is supported according to the priority.

> **Note** From the Cisco IOS XR, Release 6.3.1 onwards, you are able to enter the Flexible CLI config group definition, **apply-group** and **exclude-group** command in any order as long as the entire commit has all the group definitions needed.

Apply groups priority inheritance helps flexible configuration groups handle common configuration statements between groups. When multiple configuration groups have common configuration statements, the inheritance priority is configuration statements present in inner groups have precedence over configuration statements present in outer groups. Tiebreaker is determined by the system order (lexicographical) of the regular expressions. User defined order of commands are not accepted.

For example, a configuration statement in configuration group ONE has precedence over any other group. A configuration statement in configuration group SEVEN is used only if it is not contained in any other group. Within a configuration group, inheritance priority is lengthiest match.

```
apply-group SIX SEVEN
    router ospf 0
    apply-group FOUR FIVE
   area 0
    apply-group THREE
    interface GigabitEthernet 0/0/0/0
       apply-group ONE TWO
  !
 !
!
```

The above example states two scenarios. Inner most group (**apply-group ONE TWO**) has the highest priority. Case 1

In the first scenario it shows which group gets the first priority. The example states which group is applied between different configuration groups (different groups- nothing in common between them). While applying the group one (ONE TWO), all the seven groups that matches to the interface `interface GigabitEthernet 0/0/0/0` will be applied.

Case 2

In the case when all these groups (mentioned above) have same (common) configuration, group one will be active. The `apply-group ONE TWO` will be active. If group ONE is deleted then group TWO will be active.

Regular Expressions in Configuration Groups

Regular expressions are used in configuration groups to make them widely applicable. Portable Operating System Interface for UNIX (POSIX) 1003.2 regular expressions are supported in the names of configuration statements. Single quotes must be used to delimit a regular expression.

For general information regarding regular expressions, refer to the *Understanding Regular Expressions, Special Characters, and Patterns* module in the *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*.

Note Not all POSIX regular expressions are supported. Refer to Flexible Configuration Restrictions, on page 66 for more information.

Regular Expressions for Interface Identifiers

Configuration groups do not accept exact interface identifiers. You must use a regular expression to identify a group of interfaces that are applicable to the configuration group. The regular expression '.*' is not allowed. You must begin the regular expression for an interface identifier with an unambiguous word, followed by the regular expression. For example, to configure Gigabit Ethernet interfaces, use the regular expression 'GigabitEthernet.*'.

To display a list of available interface types for your router configuration, enter **interface ?** at the configuration group prompt:

```
RP/0/RSP0/CPU0:router(config-GRP)# interface ?

  ATM              'RegExp': ATM Network Interface(s)
  BVI              'RegExp': Bridge-Group Virtual Interface
  Bundle-Ether     'RegExp': Aggregated Ethernet interface(s)
  Bundle-POS       'RegExp': Aggregated POS interface(s)
  GigabitEthernet  'RegExp': GigabitEthernet/IEEE 802.3 interface(s)
  IMA              'RegExp': ATM Network Interface(s)
  Loopback         'RegExp': Loopback interface(s)
  MgmtEth          'RegExp': Ethernet/IEEE 802.3 interface(s)
  Multilink        'RegExp': Multilink network interface(s)
  Null             'RegExp': Null interface
  POS              'RegExp': Packet over SONET/SDH network interface(s)
  PW-Ether         'RegExp': PWHE Ethernet Interface
  PW-IW            'RegExp': PWHE VC11 IP Interworking Interface
  Serial           'RegExp': Serial network interface(s)
  tunnel-ip        'RegExp': GRE/IPinIP Tunnel Interface(s)
  tunnel-mte       'RegExp': MPLS Traffic Engineering P2MP Tunnel interface(s)
  tunnel-te        'RegExp': MPLS Traffic Engineering Tunnel interface(s)
  tunnel-tp        'RegExp': MPLS Transport Protocol Tunnel interface
```

Note Although you are required to enter only enough characters for the interface type to be unique, it is recommended that you enter the entire phrase. All interface types used in regular expressions are case-sensitive.

To specify a subinterface, prefix the expression with the characters \. (backslash period). For example, use `interface 'GigabitEthernet.*\..*'` to configure all Gigabit Ethernet subinterfaces.

You can specify Layer 2 transport interfaces or point-to-point interfaces as shown in these examples:

```
group g-l2t
  interface 'Gi.*\..*' l2transport
.
.
end-group
group g-ptp
  interface 'Gi.*\..*' point-to-point
.
.
end-group
```

Regular Expressions for an OSPF Configuration

Exact router process names and OSPF areas cannot be used. You must use a regular expression to specify a process name or group of OSPF areas. To specify that the OSFP area can be either a scalar value or an IP address, use the regular expression '.*', as in this example:

```
group g-ospf
router ospf '.*'
area '.*'
mtu-ignore enable
!
!
```

```
end-group
```

To specify that the OSPF area must be an IP address, use the expression '\.' as in this example:

```
group g-ospf-ipaddress
router ospf '.*\..*\..*\..*'
area '.*'
passive enable
!
!
 end-group
```

To specify that the OSPF area must be a scalar value, use the expression '1.*', as in this example:

```
group g-ospf-match-number
router ospf '.*'
area '1.*'
passive enable
!
!
end-group
```

Regular Expressions for a BGP AS

Exact BGP AS values cannot be used in configuration groups. Use a regular expression to specify either AS plain format, or AS dot format as in the format X.Y. To match AS plain format instances, use a simple regular expression. To match AS dot format instances, use two regular expressions separated by a dot, as shown in this example:

```
group g-bgp
router bgp '*'.'*'
address-family ipv4 unicast
!
!
end-group
```

Regular Expressions for ANCP

Exact Access Node Control Protocol (ANCP) sender-name identifiers cannot be used in configuration groups. Because the sender name argument can be either an IP address or a MAC address, you must specify in the regular expression which one is being used. Specify an IP address as `'.*\..*\..*\..*'`; specify a MAC address as `'.*\..*\..*'`.

Resolving to a Uniform Type

Regular expressions must resolve to a uniform type. This is an example of an illegal regular expression:

```
group g-invalid
 interface '.*'
  bundle port-priority 10
 !
interface '.*Ethernet.*'
  bundle port-priority 10
 !
```

```
end-group
```

In this example, the **bundle** command is supported for interface type GigabitEthernet but not for interface type ‘FastEthernet’. The regular expressions ‘.*’ and ‘.*Ethernet.*’ match both GigabitEthernet and FastEthernet types. Because the **bundle** command is not applicable to both these interface types, they do not resolve to a uniform type and therefore the system does not allow this configuration.

Note If the system cannot determine from the regular expression what the configuration should be, the expression is not considered valid.

Note The regular expression ‘.*’ is not allowed when referring to an interface identifier. You must begin the regular expression for an interface identifier with an unambiguous word, followed by the regular expression. Refer to *Regular Expressions for Interface Identifiers* in this section for more information.

Overlapping Regular Expressions

Regular expressions are used in names of configuration statements within a configuration group. This permits inheritance by the configuration when applied to matching names. Single quotes are used to delimit the regular expression. Overlapping regular expression within a configuration group for the same configuration is permitted.

The example, given below, illustrates the process of creating and applying multiple configuration groups:

```
RP/0/RSP0/CPU0:router(config)#group FB_flexi_snmp
RP/0/RSP0/CPU0:router(config-GRP)# snmp-server vrf '.*'
RP/0/RSP0/CPU0:router(config-GRP-snmp-vrf)# host 1.1.1.1 traps version 2c group_1
RP/0/RSP0/CPU0:router(config-GRP-snmp-vrf)# host 1.1.1.1 informs version 2c group_1
RP/0/RSP0/CPU0:router(config-GRP-snmp-vrf)# context group_1

RP/0/RSP0/CPU0:router(config-GRP-snmp-vrf)#
RP/0/RSP0/CPU0:router(config-GRP-snmp-vrf)#commit

RP/0/RSP0/CPU0:router(config-GRP-snmp-vrf)#root
RP/0/RSP0/CPU0:router(config)#
RP/0/RSP0/CPU0:router(config)#snmp-server vrf vrf1
RP/0/RSP0/CPU0:router(config-snmp-vrf)#snmp-server vrf vrf10
RP/0/RSP0/CPU0:router(config-snmp-vrf)#!
RP/0/RSP0/CPU0:router(config-snmp-vrf)#snmp-server vrf vrf100
RP/0/RSP0/CPU0:router(config-snmp-vrf)#
RP/0/RSP0/CPU0:router(config-snmp-vrf)#commit

RP/0/RSP0/CPU0:router(config-snmp-vrf)#root
RP/0/RSP0/CPU0:router(config)#
RP/0/RSP0/CPU0:router(config)#apply-group FB_flexi_snmp
RP/0/RSP0/CPU0:router(config)#do sh running-config group
group FB_flexi_snmp
 snmp-server vrf '.*'
  host 1.1.1.1 traps version 2c group_1
  host 1.1.1.1 informs version 2c group_1
  context group_1
 !
end-group
apply-group FB_flexi_snmp
```

```
snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
RP/0/0/CPU0:ios#show running-config inheritance detail


group FB_flexi_snmp
 snmp-server vrf '.*'
  host 1.1.1.1 traps version 2c group_1
  host 1.1.1.1 informs version 2c group_1
  context group_1
 !
end-group
snmp-server vrf vrf1
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 traps version 2c group_1
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 informs version 2c group_1
 ## Inherited from group FB_flexi_snmp
 context group_1
!
snmp-server vrf vrf10
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 traps version 2c group_1
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 informs version 2c group_1
 ## Inherited from group FB_flexi_snmp
 context group_1
!
snmp-server vrf vrf100
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 traps version 2c group_1
 ## Inherited from group FB_flexi_snmp
 host 1.1.1.1 informs version 2c group_1
 ## Inherited from group FB_flexi_snmp
 context group_1
```

The example given below demonstrates the regular expression. In this example `snmp-server vrf '.*’` and `snmp-server vrf '[\w]+` are two different regular expressions.

```
group FB_flexi_snmp
snmp-server vrf '.*’
host 1.1.1.1 traps version 2c group_1
host 1.1.1.1 informs version 2c group_1
context group_1
!
snmp-server vrf '[\w]+’
host 2.2.2.2 traps version 2c group_2
host 2.2.2.2 informs version 2c group_2
context group_2
!
end-group
```

This individual regular expression gets combined to all the three expressions - `snmp-server vrf vrf1`, `snmp-server vrf vrf10` and `snmp-server vrf vrf100` as given below.

```
apply-group FB_flexi_snmp
snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
```

In a configuration group, there can be instances of regular expressions overlap. In such cases, the regular expression with the highest priority is activated and inherited, when applied. It has that regular expression, which comes first in the lexicographic order that has the highest priority.

The following example shows how to use overlapping regular expressions and how the expression with higher priority is applied:

```
 group FB_flexi_snmp

 snmp-server vrf '.*'

  host 1.1.1.1 traps version 2c group_1

  host 1.1.1.1 informs version 2c group_1

  context group_1

 !

 snmp-server vrf '[\w]+'

  host 2.2.2.2 traps version 2c group_2
  host 2.2.2.2 informs version 2c group_2
  context group_2
 !
end-group
```

The expression shown below has the highest priority:

```
  group FB_flexi_snmp

  snmp-server vrf '.*'

  host 1.1.1.1 traps version 2c group_1

  host 1.1.1.1 informs version 2c group_1

  context group_1
```

The examples given above, show two different regular expression `snmp-server vrf '.*'` and `snmp-server vrf '[\w]+'`.

The expression below, shows how these two expressions get merged together:

```
 apply-group FB_flexi_snmp
```

```
snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
```

Any change in a regular expression with lower priority will not affect the inheritance.

Any changes made to an existing regular expression, which is of less (non-top) priority, it will not have any effect on the inheritance.

```
 snmp-server vrf '[\w]+'

 host 2.2.2.2 traps version 2c group_2
 host 2.2.2.2 informs version 2c group_2
 context group_2
```

The expression with the higher priority gets inherited, as shown below:

```
group FB_flexi_snmp

snmp-server vrf '.*'

 host 1.1.1.1 traps version 2c group_1

 host 1.1.1.1 informs version 2c group_1

 context group_1
```

Apply Groups Priority Inheritance

Priority governs inheritance.

> **Note** From the Release 6.3.1 onwards, you are able to enter the Flexible CLI config group definition, **apply-group** and **exclude-group** command in any order as long as the entire commit has all the group definitions needed.

Apply groups priority inheritance helps flexible configuration groups to handle common configuration statements between groups. When multiple configuration groups have common configuration statements, the inheritance priority is such that the configuration statements present in inner groups have precedence over those configuration statements present in outer groups. In case of tiebreakers, the priority is assigned in accordance to the lexicographical order of regular expressions. User defined order of commands are not accepted.

For example, a configuration statement in configuration group ONE has precedence over another group. A configuration statement in configuration group SEVEN is used only if it does not exist in any other group. Within a configuration group, inheritance priority is the longest match.

```
apply-group SIX SEVEN
    router ospf 0
    apply-group FOUR FIVE
   area 0
    apply-group THREE
    interface GigabitEthernet 0/0/0/0
       apply-group ONE TWO
```

```
  !
 !
!
```

The above example shows two scenarios. The inner most group (**apply-group ONE TWO**) has the highest priority. Case 1

The first scenario shows which group gets the priority. The example states which group is applied between different configuration groups (different groups with nothing in common). While applying group one (ONE TWO), all the seven groups matches the interface `interface GigabitEthernet 0/0/0/0`- is applied.

Case 2

Here, when all have the same (common) configuration, group one will be active. That is `apply-group ONE TWO` is active. If group ONE is deleted, then group TWO will be active.

Configuration Examples Using Regular Expressions

Configuration Group with Regular Expression: Example

This example shows the definition of a configuration group for configuring Gigabit Ethernet interfaces with ISIS routing parameters, using regular expressions for the exact interface:

```
RP/0/RSP0/CPU0:router(config)# group g-isis-gige
RP/0/RSP0/CPU0:router(config-GRP)# router isis '.*'
RP/0/RSP0/CPU0:router(config-GRP-isis)# interface 'GigabitEthernet.*'
RP/0/RSP0/CPU0:router(config-GRP-isis-if)# lsp-interval 20
RP/0/RSP0/CPU0:router(config-GRP-isis-if)# hello-interval 40
RP/0/RSP0/CPU0:router(config-GRP-isis-if)# address-family ipv4 unicast
RP/0/RSP0/CPU0:router(config-GRP-isis-if-af)# metric 10
RP/0/RSP0/CPU0:router(config-GRP-isis-if-af)# end-group
RP/0/RSP0/CPU0:router(config)#
```

To illustrate the use of this configuration group, assume that you want to configure these Gigabit Ethernet interfaces with the ISIS routing parameters:

```
router isis green
 interface GigabitEthernet0/0/0/0
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
   metric 10
  !
 !
 interface GigabitEthernet0/0/0/1
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
   metric 10
  !
 !
 interface GigabitEthernet0/0/0/2
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
   metric 10
  !
```

```
 !
 interface GigabitEthernet0/0/0/3
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
   metric 10
  !
 !
!
```

There are three possible ways to use the configuration group to configure these interfaces. The first is by applying the group within the interface configuration, as shown here:

```
router isis green
 interface GigabitEthernet0/0/0/0
  apply-group g-isis-gige
  !
 !
 interface GigabitEthernet0/0/0/1
  apply-group g-isis-gige
  !
 !
 interface GigabitEthernet0/0/0/2
  apply-group g-isis-gige
  !
 !
 interface GigabitEthernet0/0/0/3
  apply-group g-isis-gige
  !
 !
```

In this situation, only the interfaces to which you apply the configuration group inherit the configuration.

The second way to configure these interfaces using the configuration group is to apply the configuration group within the **router isis** configuration, as shown here:

```
router isis green
 apply-group g-isis-gige
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
 interface GigabitEthernet0/0/0/3
 !
!
```

In this way, any other Gigabit Ethernet interfaces that you configure in the ISIS green configuration also inherit these configurations.

The third way to configure these interfaces using the configuration group is to apply the group at the global level as shown here:

```
apply-group g-isis-gige
router isis green
 interface GigabitEthernet0/0/0/0
```

```
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
 interface GigabitEthernet0/0/0/3
 !
!
```

In this example, the configuration of the group is applied to all Gigabit Ethernet interfaces configured for ISIS.

Configuration Group Inheritance with Regular Expressions: Example

Local Configuration Has Precedence Over Configuration Group

An explicit configuration takes precedence over a configuration applied from a configuration group. For example, assume that this configuration is running on the router:

```
router ospf 100
 packet-size 1000
!
```

You configure this configuration group, apply it, and commit it to the configuration.

```
RP/0/RSP0/CPU0:router(config)# group g-ospf
RP/0/RSP0/CPU0:router(config-GRP)# router ospf '.*'
RP/0/RSP0/CPU0:router(config-GRP-ospf)# nsf cisco
RP/0/RSP0/CPU0:router(config-GRP-ospf)# packet-size 3000
RP/0/RSP0/CPU0:router(config-GRP-ospf)# end-group

RP/0/RSP0/CPU0:router(config)# apply-group g-ospf
```

The result is effectively this configuration:

```
router ospf 100
 packet-size 1000
 nsf cisco
```

Note that `packet-size 3000` is not inherited from the configuration group because the explicit local configuration has precedence.

Compatible Configuration Is Inherited

The configuration in the configuration group must match the configuration on the router to be inherited. If the configuration does not match, it is not inherited. For example, assume that this configuration is running on the router:

```
router ospf 100
 auto-cost disable
!
```

You configure this configuration and commit it to the configuration.

```
RP/0/RSP0/CPU0:router(config)# group g-ospf
RP/0/RSP0/CPU0:router(config-GRP)# router ospf '.*'
RP/0/RSP0/CPU0:router(config-GRP-ospf)# area '.*'
RP/0/RSP0/CPU0:router(config-GRP-ospf-ar)# packet-size 2000
RP/0/RSP0/CPU0:router(config-GRP-ospf)# end-group

RP/0/RSP0/CPU0:router(config)# apply-group g-ospf

RP/0/RSP0/CPU0:router(config)# router ospf 200
RP/0/RSP0/CPU0:router(config-ospf)# area 1
```

The result is effectively this configuration:

```
router ospf 100
  auto-cost disable

router ospf 200
  area 1
  packet-size 2000
```

The packet size is inherited by the ospf 200 configuration, but not by the ospf 100 configuration because the area is not configured.

Layer 2 Transport Configuration Group: Example

This example shows how to configure and apply a configuration group with Layer 2 transport subinterfaces:

```
RP/0/RSP0/CPU0:router(config)# group g-l2trans-if
RP/0/RSP0/CPU0:router(config-GRP)# interface 'TenGigE.*\..*' l2transport
RP/0/RSP0/CPU0:router(config-GRP)# mtu 1514
RP/0/RSP0/CPU0:router(config-GRP)# end-group

RP/0/RSP0/CPU0:router(config)# interface TenGigE0/0/0/0.1 l2transport
RP/0/RSP0/CPU0:router(config-if)# apply-group g-l2trans-if
```

When this configuration is committed, the Ten Gigabit Ethernet interface 0/0/0/0.1 inherits the 1514 MTU value. This is the output displayed from the **show running-config inheritence** command for the Ten Gigabit Ethernet interface:

```
interface TenGigE0/0/0/0.1 l2transport
 ## Inherited from group g-l2trans-if
 mtu 1514
!
```

Configuration Group Precedence: Example

When similar configuration statements are contained in multiple configuration groups, groups applied in inner configuration modes take precedence over groups applied in outer modes. This example shows two configuration groups that configure different cost values for OSPF.

```
RP/0/RSP0/CPU0:router(config)# group g-ospf2
RP/0/RSP0/CPU0:router(config-GRP)# router ospf '.*'
RP/0/RSP0/CPU0:router(config-GRP-ospf)# area '.*'
RP/0/RSP0/CPU0:router(config-GRP-ospf-ar)# cost 2
RP/0/RSP0/CPU0:router(config-GRP-ospf-ar)# end-group

RP/0/RSP0/CPU0:router(config)# group g-ospf100
RP/0/RSP0/CPU0:router(config-GRP)# router ospf '.*'
RP/0/RSP0/CPU0:router(config-GRP-ospf)# area '.*'
RP/0/RSP0/CPU0:router(config-GRP-ospf-ar)# cost 100
RP/0/RSP0/CPU0:router(config-GRP-ospf-ar)# end-group
```

If these configuration groups are applied as follows, the cost 2 specified in g-ospf2 is inherited by OSPF area 0 because the group is applied in a more inner configuration mode. In this case, the configuration in group g-ospf100 is ignored.

```
RP/0/RSP0/CPU0:router(config)# router ospf 0
RP/0/RSP0/CPU0:router(config-ospf)# apply-group g-ospf100
RP/0/RSP0/CPU0:router(config-ospf)# area 0
RP/0/RSP0/CPU0:router(config-ospf-ar)# apply-group g-ospf2
```

Changes to Configuration Group are Automatically Inherited: Example

When you make changes to a configuration group that is committed and applied to your router configuration, the changes are automatically inherited by the router configuration. For example, assume that this configuration is committed:

```
group g-interface-mtu
 interface ‘POS.*’
  mtu 1500
 !
end-group

interface POS0/4/1/0
 apply-group g-interface-mtu
!
```

Now you change the configuration group as in this example:

```
RP/0/RSP0/CPU0:router(config)# group g-interface-mtu
RP/0/RSP0/CPU0:router(config-GRP)# interface 'POS.*'
RP/0/RSP0/CPU0:router(config-GRP-if)# mtu 2000
RP/0/RSP0/CPU0:router(config-GRP-if)# end-group
```

When this configuration group is committed, the MTU configuration for interface POS0/4/1/0 is automatically updated to 2000.

Configuration Examples for Flexible CLI Configuration

Basic Flexible CLI Configuration: Example

This example shows that the Media Access Control (MAC) accounting configuration from the gd21 configuration group is applied to all Gigabit Ethernet interfaces in slot 2, ports 1 to 9.

1. Configure the configuration group that configures MAC accounting:

```
RP/0/RSP0/CPU0:router# show running group gd21

group gd21
interface 'GigabitEthernet0/0/0/2[1-9]'
description general interface inheritance check
load-interval 30
mac-accounting ingress
mac-accounting egress
!
end-group
```

2. Check that the corresponding apply-group is configured in global configuration or somewhere in the hierarchy:

```
RP/0/RSP0/CPU0:router# show running | in apply-group gd21

Building configuration...
apply-group gd21
```

3. Check the concise local view of the configuration of some of the interfaces:

```
RP/0/RSP0/CPU0:router# show running interface

interface GigabiEthernet0/0/0/21
!
interface GigabitEthernet0/0/0/22
!
```

4. Verify that the match and inheritance occur on these interfaces:

```
RP/0/RSP0/CPU0:router# show running inheritance interface

interface GigabitEthernet0/0/0/21
## Inherited from group gd21
description general interface inheritance check
## Inherited from group gd21
load-interval 30
## Inherited from group gd21
mac-accounting ingress
## Inherited from group gd21
mac-accounting egress
!
Interface GigabitEthernet0/0/0/22
## Inherited from group gd21
description general interface inheritance check
```

```
## Inherited from group gd21
load-interval 30
## Inherited from group gd21
mac-accounting ingress
## Inherited from group gd21
mac-accounting egress
!
!
```

5. Verify that the inherited configuration actually takes effect:

```
RP/0/RSP0/CPU0:router# show mac gigabitEthernet0/0/0/21

GigabitEthernet0/0/0/21
  Input (96 free)
    6c9c.ed35.90fd:  1271 packets, 98426 bytes
        Total: 1271 packets, 98426 bytes
  Output (96 free)
    6c9c.ed35.90fd:   774 packets, 63265 bytes
        Total: 774 packets, 63264 bytes
```

Interface MTU Settings for Different Interface Types: Example

This example shows that an MTU value is configured on different interface types.

1. Configure an interface MTU configuration group and apply this group:

```
RP/0/RSP0/CPU0:router# show running group l2tr

group l2tr
interface 'GigabitEthernet0/0/0/3.*'
mtu 1500
!
interface 'GigabitEthernet0/0/0/9\..*'
mtu 1400
!
interface 'GigabitEthernet0/0/0/9\..*' l2transport
mtu 1400
!
end-group

RP/0/RSP0/CPU0:router# show running | inc apply-group

Building configuration...

apply-group l2tr
```

2. Check the concise view and the inheritance view of the various interfaces:

```
RP/0/RSP0/CPU0:router# show running interface gigabitEthernet0/0/0/30

interface GigabitEthernet0/0/0/30
!
RP/0/RSP0/CPU0:router# show running inheritance interface gigabitEthernet0/0/0/30

interface GigabitEthernet0/0/0/30
```

```
## Inherited from group l2tr
mtu 1500
!

RP/0/RSP0/CPU0:router# show running interface gigabitEthernet0/0/0/9.800

interface GigabitEthernet0/0/0/9.800
  encapsulation dot1q 800
!

RP/0/RSP0/CPU0:router# show running inheritance interface gigabitEthernet0/0/0/9.800

interface GigabitEthernet0/0/0/9.800
## Inherited from group l2tr
mtu 1400
encapsulation dot1q800
!

RP/0/RSP0/CPU0:router# show running interface gigabitEthernet0/0/0/9.250

interface GigabitEthernet0/0/0/9.250 l2transport
  encapsulation dot1q 250
!

RP/0/RSP0/CPU0:router# show running inheritance interface gigabitEthernet0/0/0/9.800

interface GigabitEthernet0/0/0/9.250 l2transport
encapsulation dot1q250
## Inherited from group l2tr
mtu 1400
!
```

3. Verify that the correct values from the group do take effect:

```
RP/0/RSP0/CPU0:router# show interface gigabitEthernet 0/0/0/30

GigabitEthernet0/0/0/30 is down, line protocol is down
  Interface state transitions: 0
  Hardware is GigabitEthernet, address is 0026.9824.ee56 (bia 0026.9824.ee56)
  Internet address is Unknown
  MTU 1500 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, link type is force-up
  output flow control is off, input flow control is off
  loopback not set,
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out

RP/0/RSP0/CPU0:router# show interface gigabitEthernet 0/0/0/9.801
```

```
GigabitEthernet0/0/0/9.801 is up, line protocol is up
  Interface state transitions: 1
  Hardware is VLAN sub-interface(s), address is 0026.9824.ee41
  Internet address is Unknown
  MTU 1400 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN, VLAN Id 801,  loopback not set,
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets


RP/0/RSP0/CPU0:router# show interface gigabitEthernet 0/0/0/9.250

GigabitEthernet0/0/0/9.250 is up, line protocol is up
  Interface state transitions: 1
  Hardware is VLAN sub-interface(s), address is 0026.9824.ee41
  Layer 2 Transport Mode
  MTU 1400 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability Unknown, txload Unknown, rxload Unknown
  Encapsulation 802.1Q Virtual LAN,
    Outer Match: Dot1Q VLAN 250
    Ethertype Any, MAC Match src any, dest any
  loopback not set,
  Last input never, output never
  Last clearing of "show interface" counters never
     0 packets input, 0 bytes
     0 input drops, 0 queue drops, 0 input errors
     0 packets output, 0 bytes

     0 output drops, 0 queue drops, 0 output errors
```

ACL Referencing: Example

This example shows how to reference access-lists on a number of interfaces using configuration groups.

1. Configure the configuration group and apply-group:

```
RP/0/RSP0/CPU0:router# show running group acref

group acref
 interface 'GigabitEthernet0/0/0/3.*'
  ipv4 access-group adem ingress
  ipv4 access-group adem egress
 !
end-group

RP/0/RSP0/CPU0:router# show running | inc apply-group

Building configuration...
```

```
apply-group isis l2tr isis2 mpp bundle1 acref
```

2. Check the concise and inheritance view of the matching configurations:

```
RP/0/RSP0/CPU0:router# show running interface gigabitEthernet 0/0/0/30

interface GigabitEthernet0/0/0/30
!

RP/0/RSP0/CPU0:router# show running inheritance interface GigabitEthernet 0/0/0/30

interface GigabitEthernet0/0/0/30
 ## Inherited from group l2tr
 mtu 1500
 ## Inherited from group acref
 ipv4 access-group adem ingress
 ## Inherited from group acref
 ipv4 access-group adem egress
!

RP/0/RSP0/CPU0:router# show running interface gigabitEthernet 0/0/0/31

interface GigabitEthernet0/0/0/31
!

RP/0/RSP0/CPU0:router# show running inheritance interface GigabitEthernet 0/0/0/31

interface GigabitEthernet0/0/0/31
 ## Inherited from group l2tr
 mtu 1500
 ## Inherited from group acref
 ipv4 access-group adem ingress
 ## Inherited from group acref
 ipv4 access-group adem egress
```

3. Check that the ACL group configuration actually got configured by using a traffic generator and watching that denied traffic is dropped.

Local Configuration Takes Precedence: Example

This example illustrates that local configurations take precedence when there is a discrepancy between a local configuration and the configuration inherited from a configuration group.

1. Configure a local configuration in a configuration submode with an access list:

```
RP/0/RSP0/CPU0:router# show running interface gigabitEthernet 0/0/0/39

interface GigabitEthernet0/0/0/39
 ipv4 access-group smany ingress
 ipv4 access-group smany egress
!

RP/0/RSP0/CPU0:router# show running interface gigabitEthernet 0/0/0/38

interface GigabitEthernet0/0/0/38
!
```

```
RP/0/RSP0/CPU0:router# show running ipv4 access-list smany

ipv4 access-list smany
 10 permit ipv4 any any
!

RP/0/RSP0/CPU0:router# show running ipv4 access-list adem

ipv4 access-list adem
 10 permit ipv4 21.0.0.0 0.255.255.255 host 55.55.55.55
 20 deny ipv4 any any
!
```

2. Configure and apply the access list group configuration:

```
RP/0/RSP0/CPU0:router# show running group acref

group acref
 interface 'GigabitEthernet0/0/0/3.*'
  ipv4 access-group adem ingress
  ipv4 access-group adem egress
 !
end-group

RP/0/RSP0/CPU0:router# show running | inc apply-group

Building configuration...
apply-group isis l2tr isis2 mpp bundle1 acref
```

3. Check the concise and inheritance views for the matching interface where the access list reference is configured locally:

```
RP/0/RSP0/CPU0:router# show running interface gigabitEthernet 0/0/0/39

interface GigabitEthernet0/0/0/39
 ipv4 access-group smany ingress
 ipv4 access-group smany egress
!

RP/0/RSP0/CPU0:router# show running inheritance interface gigabitEthernet 0/0/0/39

interface GigabitEthernet0/0/0/39
 ## Inherited from group l2tr
 mtu 1500
 ipv4 access-group smany ingress
 ipv4 access-group smany egress     << no config inherited, local config prioritized
!

RP/0/RSP0/CPU0:router# show running interface gigabitEthernet 0/0/0/38

interface GigabitEthernet0/0/0/38
!

RP/0/RSP0/CPU0:router# show running inheritance interface gigabitEthernet 0/0/0/38

interface GigabitEthernet0/0/0/38
 ## Inherited from group l2tr
 mtu 1500
 ## Inherited from group acref
 ipv4 access-group adem ingress
 ## Inherited from group acref
```

```
 ipv4 access-group adem egress
!
```

4. Use a traffic generator to verify that the traffic pattern for interface GigabitEthernet0/0/0/39 gets acted on by the access list in the local configuration (smany) and not according to the inherited referenced access list (adem).

ISIS Hierarchical Configuration: Example

This example illustrates inheritance and priority handling with two ISIS groups using an ISIS configuration.

1. Configure the local ISIS configuration:

```
RP/0/RSP0/CPU0:router# show running router isis

router isis vink
 net 49.0011.2222.2222.2222.00
 address-family ipv4 unicast
  mpls traffic-eng level-1-2
  mpls traffic-eng router-id Loopback0
  redistribute connected
 !
 interface Bundle-Ether1
  address-family ipv4 unicast
  !
 !
 interface Bundle-Ether2
 !
 interface Loopback0
 !
 interface TenGigE0/2/0/0.3521
  address-family ipv4 unicast
  !
 !
 interface TenGigE0/2/0/0.3522
  address-family ipv4 unicast
  !
 !
 interface TenGigE0/2/0/0.3523
  address-family ipv4 unicast
  !
 !
 interface TenGigE0/2/0/0.3524
  address-family ipv4 unicast
  !
 !
 interface TenGigE0/2/0/0.3525
  address-family ipv4 unicast
  !
 !
 interface TenGigE0/2/0/0.3526
 !
 interface TenGigE0/2/0/0.3527
 !
 interface TenGigE0/2/0/0.3528
 !
 interface TenGigE0/2/0/1
```

```
  address-family ipv4 unicast
  !
 !
!
```

2. Configure two ISIS groups and apply these to the configuration:

```
RP/0/RSP0/CPU0:router# show running group isis

group isis
 router isis '.*'
  address-family ipv4 unicast
   mpls traffic-eng level-1-2
   mpls traffic-eng router-id Loopback0
   redistribute connected
   redistribute ospf 1 level-1-2
  !
  interface 'TenGig.*'
   lsp-interval 40
   hello-interval 15
   address-family ipv4 unicast
    metric 50
   !
  !
  interface 'Bundle-Ether.*'
   address-family ipv4 unicast
    metric 55
   !
  !
 !
end-group

RP/0/RSP0/CPU0:router# show running group isis2

group isis2
 router isis '.*'
 !
 router isis '^(vink)'
  address-family ipv4 unicast
  !
  interface '(^Ten)Gig.*'
  !
  interface '^(Ten)Gig.*'
   address-family ipv4 unicast
    metric 66
   !
  !
 !
end-group

RP/0/RSP0/CPU0:router# show running | inc apply-group

Building configuration...

apply-group isis  l2tr  isis2  mpp  bundle1  acref
```

3. Check the inheritance view of the ISIS configuration:

```
RP/0/RSP0/CPU0:router# show running inheritance router isis

router isis vink
```

```
net 49.0011.2222.2222.2222.00
address-family ipv4 unicast
 mpls traffic-eng level-1-2
 mpls traffic-eng router-id Loopback0
 redistribute connected
 ## Inherited from group isis
 redistribute ospf 1 level-1-2
!
interface Bundle-Ether1
 address-family ipv4 unicast
  ## Inherited from group isis
  metric 55
 !
!
interface Bundle-Ether2
 ## Inherited from group isis
 address-family ipv4 unicast
  ## Inherited from group isis
  metric 55
 !
!
interface Loopback0
!
interface TenGigE0/2/0/0.3521
 ## Inherited from group isis
 lsp-interval 40
 ## Inherited from group isis
 hello-interval 15
 address-family ipv4 unicast
  ## Inherited from group isis
  metric 50
 !
!
interface TenGigE0/2/0/0.3522
 ## Inherited from group isis
 lsp-interval 40
 ## Inherited from group isis
 hello-interval 15
 address-family ipv4 unicast
  ## Inherited from group isis
  metric 50
 !
!
interface TenGigE0/2/0/0.3523
 ## Inherited from group isis
 lsp-interval 40
 ## Inherited from group isis
 hello-interval 15
 address-family ipv4 unicast
  ## Inherited from group isis
  metric 50
 !
!
interface TenGigE0/2/0/0.3524
 ## Inherited from group isis
 lsp-interval 40
 ## Inherited from group isis
 hello-interval 15
 address-family ipv4 unicast
  ## Inherited from group isis
  metric 50
 !
!
interface TenGigE0/2/0/0.3525
```

```
  ## Inherited from group isis
  lsp-interval 40
  ## Inherited from group isis
  hello-interval 15
  address-family ipv4 unicast
   ## Inherited from group isis
   metric 50
  !
 !
 interface TenGigE0/2/0/0.3526
  ## Inherited from group isis
  lsp-interval 40
  ## Inherited from group isis
  hello-interval 15
  ## Inherited from group isis
  address-family ipv4 unicast
   ## Inherited from group isis
   metric 50
  !
 !
 interface TenGigE0/2/0/0.3527
  ## Inherited from group isis
  lsp-interval 40
  ## Inherited from group isis
  hello-interval 15
  ## Inherited from group isis
  address-family ipv4 unicast
   ## Inherited from group isis
   metric 50
  !
 !
 interface TenGigE0/2/0/0.3528
  ## Inherited from group isis
  lsp-interval 40
  ## Inherited from group isis
  hello-interval 15
  ## Inherited from group isis
  address-family ipv4 unicast
   ## Inherited from group isis
   metric 50
  !
 !
 interface TenGigE0/2/0/1
  ## Inherited from group isis
  lsp-interval 40
  ## Inherited from group isis
  hello-interval 15
  address-family ipv4 unicast
   ## Inherited from group isis
   metric 50
  !
 !
!
```

4. Verify the actual functionality:

```
RP/0/RSP0/CPU0:router# show isis interface TenGigE0/2/0/0.3528 | inc Metric

Metric (L1/L2):         50/50
```

OSPF Hierarchy: Example

This example illustrates hierarchical inheritance and priority. The configuration that is lower in hierarchy gets the highest priority.

1. Configure a local OSPF configuration:

```
RP/0/RSP0/CPU0:router# show running router ospf

router ospf 1
 apply-group go-c
 nsr
 router-id 121.121.121.121
 nsf cisco
 redistribute connected
 address-family ipv4 unicast
 area 0
  apply-group go-b
  interface GigabitEthernet0/0/0/0
   apply-group go-a
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/3
  !
  interface GigabitEthernet0/0/0/4
  !
  interface GigabitEthernet0/0/0/21
   bfd minimum-interval 100
   bfd fast-detect
   bfd multiplier 3
  !
  interface TenGigE0/2/0/0.3891
  !
  interface TenGigE0/2/0/0.3892
  !
  interface TenGigE0/2/0/0.3893
  !
  interface TenGigE0/2/0/0.3894
  !
 !
!
router ospf 100
!
router ospf 1000
!
router ospf 1001
!
```

2. Configure a configuration group and apply it in a configuration submode:

```
RP/0/RSP0/CPU0:router# show running group go-a

group go-a
 router ospf '.*'
  area '.*'
   interface 'Gig.*'
    cost 200
   !
  !
 !
```

```
end-group

RP/0/RSP0/CPU0:router# show running group go-b

group go-b
 router ospf '.*'
  area '.*'
   interface 'Gig.*'
    cost 250
   !
  !
 !
end-group

RP/0/RSP0/CPU0:router# show running group go-c

group go-c
 router ospf '.*'
  area '.*'
   interface 'Gig.*'
    cost 300
   !
  !
 !

end-group
```

3. Check the inheritance view and verify that the apply-group in the lowest configuration submode gets the highest priority:

```
RP/0/RSP0/CPU0:router# show running inheritance router ospf 1

router ospf 1
 nsr
 router-id 121.121.121.121
 nsf cisco
 redistribute connected
 address-family ipv4 unicast
 area 0
  interface GigabitEthernet0/0/0/0
   ## Inherited from group go-a
   cost 200                    << apply-group in lowest submode  gets highest priority

  !
  interface GigabitEthernet0/0/0/1
   ## Inherited from group go-b
   cost 250
  !
  interface GigabitEthernet0/0/0/3
   ## Inherited from group go-b
   cost 250
  !
  interface GigabitEthernet0/0/0/4
   ## Inherited from group go-b
   cost 250
  !
  interface GigabitEthernet0/0/0/21
   bfd minimum-interval 100
   bfd fast-detect
   bfd multiplier 3
   ## Inherited from group go-b
   cost 250
  !
```

```
  interface TenGigE0/2/0/0.3891
  !
  interface TenGigE0/2/0/0.3892
  !
  interface TenGigE0/2/0/0.3893
  !
  interface TenGigE0/2/0/0.3894
  !
 !
!
```

4. Check the functionality of the cost inheritance through the groups:

```
RP/0/RSP0/CPU0:router# show ospf 1 interface GigabitEthernet 0/0/0/0

GigabitEthernet0/0/0/0 is up, line protocol is up
  Internet Address 1.0.1.1/30, Area 0
  Process ID 1, Router ID 121.121.121.121, Network Type BROADCAST, Cost: 200
  Transmit Delay is 1 sec, State DR, Priority 1, MTU 1500, MaxPktSz 1500
  Designated Router (ID) 121.121.121.121, Interface address 1.0.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Non-Stop Forwarding (NSF) enabled
    Hello due in 00:00:02
  Index 5/5, flood queue length 0
  Next 0(0)/0(0)
  Last flood scan length is 1, maximum is 40
  Last flood scan time is 0 msec, maximum is 7 msec
  LS Ack List: current length 0, high water mark 0
  Neighbor Count is 1, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
  Multi-area interface Count is 0
```

Link Bundling Usage: Example

This example shows how to configure interface membership in a bundle link:

1. Configure the configuration groups:

```
RP/0/RSP0/CPU0:router# show running group bundle1

group bundle1
 interface 'GigabitEthernet0/1/0/1[1-6]'
  bundle id 1 mode active
 !
end-group

RP/0/RSP0/CPU0:router# show running | inc apply-group

Building configuration...

apply-group isis l2tr  isis2  mpp  bundle1
```

2. Check the local configuration:

```
RP/0/RSP0/CPU0:router# show running interface gigabitEthernet 0/1/0/11
```

```
interface GigabitEthernet0/1/0/11
!

RP/0/RSP0/CPU0:router# show running interface Bundle-Ether1

interface Bundle-Ether1
 ipv4 address 108.108.1.1 255.255.255.0
 bundle maximum-active links 10
 bundle minimum-active links 5
!
```

3. Check the inheritance configuration view:

```
RP/0/RSP0/CPU0:router# show running inheritance interface GigabitEthernet 0/1/0/11

interface GigabitEthernet0/1/0/11
 ## Inherited from group bundle1
 bundle id 1 mode active
!
```

4. Check that the inheritance configuration took effect:

```
RP/0/RSP0/CPU0:router# show interface Bundle-Ether1

Bundle-Ether1 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Aggregated Ethernet interface(s), address is 0024.f71f.4bc3
  Internet address is 108.108.1.1/24
  MTU 1514 bytes, BW 6000000 Kbit (Max: 6000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 6000Mb/s
  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
    No. of members in this bundle: 6
      GigabitEthernet0/1/0/11      Full-duplex  1000Mb/s     Active
      GigabitEthernet0/1/0/12      Full-duplex  1000Mb/s     Active
      GigabitEthernet0/1/0/13      Full-duplex  1000Mb/s     Active
      GigabitEthernet0/1/0/14      Full-duplex  1000Mb/s     Active
      GigabitEthernet0/1/0/15      Full-duplex  1000Mb/s     Active
      GigabitEthernet0/1/0/16      Full-duplex  1000Mb/s     Active
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 8000 bits/sec, 1 packets/sec
  5 minute output rate 3000 bits/sec, 1 packets/sec
     2058 packets input, 1999803 bytes, 426 total input drops
     0 drops for unrecognized upper-level protocol
     Received 1 broadcast packets, 2057 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1204 packets output, 717972 bytes, 0 total output drops
     Output 2 broadcast packets, 1202 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Replacing Configuration Elements

You can replace interface and IP address configurations, or any pattern in an existing configuration, using the **replace** {**interface**} or **replace** {**pattern**}commands in Configuration mode. These commands can be executed not only on individual interfaces or IP addresses, but also on regular expressions to replace a range of interfaces or addresses.

Use these commands to simplify configuration changes where you would normally need to copy the configuration and edit it manually. For example, when you're moving a physical connection from one interface to another, you can use the **replaceinterface** command to update your configuration to use the new interface address.

Note These commands replace every occurrence of the specified interfaces or patterns in the running configuration.

Note We recommend that you use this command after disconnecting the old interface and before connecting to the new interface.

Similarly, if your IP addressing scheme has changed (for example, a BGP neighbor address), use the **replace pattern** command to update your configuration to use the new IP address.

The following configuration examples are provided in this document:

1. Replacing interface configurations
2. Replacing IP addresses in a configuration
3. Replacing patterns using regular expressions

Replacing an Interface Configuration

The example in this section uses the following interface configurations:

```
Router# show configuration running-config
. . .
interface MgmtEth0/RSP0/CPU0/0
 shutdown
!
interface HundredGigE0/0/0/0
 description first
 ipv4 address 10.20.30.40 255.255.0.0
 shutdown
!
interface HundredGigE0/0/0/1
 shutdown
!
interface HundredGigE0/0/0/2
 description 10.20.30.40
 shutdown
!
interface HundredGigE0/0/0/3
 description 1020304050607080
 shutdown
!
interface HundredGigE0/0/0/4
```

```
 description 1.2.3.4.5.6.7.8
 shutdown
!
router ospf 10
 area 200
  interface HundredGigE0/0/0/0
   transmit-delay 5
  !
 !
!
end
```

This example shows how to replace the HundredGigE0/0/0/0 with HundredGigE0/1/0/1 using the **replace interface** *type interface-path-id* **with** *type interface-path-id* command:

```
Router(config)# replace interface HundredGigE0/0/0/0 with HundredGigE0/1/0/1
Loading.
272 bytes parsed in 1 sec (271)bytes/sec
```

Enter the **show configuration** command to display and verify the configuration changes. Then commit the changes.

```
Router(config)# show configuration
Thu May  7 21:24:29.182 UTC
Building configuration...
!! IOS XR Configuration 0.0.0
no interface HundredGigE0/0/0/0
interface HundredGigE0/1/0/1
 description first
 ipv4 address 10.20.30.40 255.255.0.0
 shutdown
!
router ospf 10
 area 200
  no interface HundredGigE0/0/0/0
  interface HundredGigE0/1/0/1
   transmit-delay 5
  !
 !
!
end

Router(config)# commit
Thu May  7 21:24:48.985 UTC
```

In the example above, you can see that every occurrence of HundredGigE0/0/0/0 is removed from the configuration (no interface HundredGigE0/0/0/0) and is replaced with HundredGigE0/1/0/1.

Replacing an IP Address in a Configuration

The example in this section uses the following configuration:

```
Router# show configuration running-config
. . .
ipv4 access-list mylist
 10 permit tcp 10.20.30.40/16 host 1.2.4.5
 20 deny ipv4 any 1.2.3.6/16
!
interface MgmtEth0/RSP0/CPU0/0
 shutdown
!
interface HundredGigE0/1/0/1
 description first
 ipv4 address 10.20.30.40 255.255.0.0
```

```
 shutdown
!
interface HundredGigE0/0/0/2
 description 10.20.30.40
 shutdown
!
route-policy temp
  if ospf-area is 10.20.30.40 or source in (2.3.4.5/20) then
    pass
  endif
end-policy
!
```

This example shows how to replace IP address 10.20.30.40 with 100.200.250.225 using the **replace pattern** *'pattern'*“pattern’ **with** *'pattern'* command:

Note

Use single quotes around the pattern.

```
Router(config)# replace pattern '10.20.30.40' with '100.200.250.225'
Loading.
443 bytes parsed in 1 sec (442)bytes/sec
```

Enter the **show configuration** command to display and verify the configuration changes. Then commit the changes.

```
Router(config)# show configuration
Thu May  7 21:45:30.170 UTC
Building configuration...
!! IOS XR Configuration 0.0.0
ipv4 access-list mylist
 no 10
 10 permit tcp 100.200.250.225/16 host 1.2.4.5
!
interface HundredGigE0/0/0/2
 no description
 description 100.200.250.225
!
interface HundredGigE0/1/0/1
 no ipv4 address 10.20.30.40 255.255.0.0
 ipv4 address 100.200.250.225 255.255.0.0
!
!
route-policy temp
  if ospf-area is 100.200.250.225 or source in (2.3.4.5/20) then
    pass
  endif
end-policy
!
end

Router(config)# commit
Thu May  7 21:46:48.985 UTC
```

In the example above, you can see that every occurrence of IP address 10.20.30.40 has been replaced with 100.200.250.225.

Replace a Pattern Using Regular Expressions

You can replace a range of interfaces or addresses using POSIX-compliant regular expressions.

Note

For information about using regular expressions, refer to the "Understanding Regular Expressions, Special Characters, and Patterns" chapter in the *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*.

The example in this section uses the following configuration:

```
Router# show configuration running-config
. . .
interface HundredGigE0/2/0/0
 ipv4 address 10.0.0.10 255.255.0.0
!
interface HundredGigE0/2/0/1
 ipv4 address 11.0.0.11 255.255.0.0
!
interface HundredGigE0/2/0/2
 ipv4 address 12.0.0.12 255.255.0.0
!
interface HundredGigE0/2/0/3
 ipv4 address 13.0.0.13 255.255.0.0
!
interface HundredGigE0/2/0/4
 ipv4 address 14.0.0.14 255.255.0.0
!
interface HundredGigE0/3/0/0
 shutdown
!
interface HundredGigE0/3/0/1
 shutdown
!
interface HundredGigE0/3/0/2
 shutdown
!
interface HundredGigE0/3/0/3
 shutdown
!
interface HundredGigE0/3/0/4
 shutdown
!
interface HundredGigE0/3/0/5
 shutdown
!
interface HundredGigE0/3/0/6
 shutdown
!
end
```

This example shows how to replace interfaces HundredGigE0/2/0/0 through HundredGigE0/2/0/4 with interfaces HundredGigE0/3/0/0 through HundredGigE0/3/0/4 using regular expressions:

```
Router(config)# replace pattern 'HundredGigE0/2/0/([0-4]*)' with 'HundredGigE0/3/0/1'
Loading.
619 bytes parsed in 1 sec (617)bytes/sec
```

Enter the **show configuration** command to display and verify the configuration changes. Then commit the changes.

```
Router(config)# show configuration
Thu May  7 22:02:09.273 UTC
Building configuration...
!! IOS XR Configuration 0.0.0
no interface HundredGigE0/2/0/0
```

```
no interface HundredGigE0/2/0/1
no interface HundredGigE0/2/0/2
no interface HundredGigE0/2/0/3
no interface HundredGigE0/2/0/4
interface HundredGigE0/3/0/0
 ipv4 address 10.0.0.10 255.255.0.0
!
interface HundredGigE0/3/0/1
 ipv4 address 11.0.0.11 255.255.0.0
!
interface HundredGigE0/3/0/2
 ipv4 address 12.0.0.12 255.255.0.0
!
interface HundredGigE0/3/0/3
 ipv4 address 13.0.0.13 255.255.0.0
!
interface HundredGigE0/3/0/4
 ipv4 address 14.0.0.14 255.255.0.0
!
End

Router(config)# commit
Thu May  7 22:05:50.015 UTC
Router(config)#
```

In the example above, you can see that the HundredGigE0/2/0/*x* interfaces are removed from the configuration (no interface HundredGigE0/2/0/*x*) and is replaced with HundredGigE0/3/0/*x*.

CHAPTER 7

Managing Router Hardware

This chapter describes the command-line interface (CLI) techniques and commands used to manage and configure the hardware components of a router running the Cisco IOS XR software.

For complete descriptions of the commands listed in this module, see Additional References, on page 164. To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 9: Feature History for Managing Router Hardware with Cisco IOS XR Software

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.9.0	No modification.
Release 6.5.2	The Cisco CPAK 100GBASE-ER4 Lite (CPAK-100G-ER4L) module is supported on Cisco IOS XR 64-bit operating system with the following line cards: • A9K-8X100G-LB-SE • A9K-8X100G-LB-TR • A9K-8X100G-SE • A9K-8X100G-TR • A9K-8X100G-CM • A99-8X100G-SE • A99-8X100G-TR • A99-8X100G-CM • A9K-4X100G-TR • A9K-4X100G-SE

This module contains the following topics:

Prerequisites for Managing Router Hardware

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

Displaying Hardware Status

This section describes how to display different types of hardware status information.

Displaying SDR Hardware Version Information

To display hardware version information for the components assigned to a secure domain router (SDR), connect to the designated shelf controller (DSC) and enter the **show diag** command in EXEC mode. The displayed information includes the card serial number and the ROMMON software version.

The syntax for the **show diag** command in EXEC mode is:

show diag [*node-id* | **details** | **summary**]

In the following example, the **show diag** command displays information for all nodes in the SDR:

```
RP/0/RSP0/CPU0:router# show diag

Mon Jun 29 00:36:41.576 PST

NODE module 0/RSP0/CPU0 :

  MAIN:  board type 0x100302
```

```
  S/N:   FOC1230803H
  Top Assy. Number:   68-3160-04
  PID:   A2K-RSP-4G-HDD=
  UDI_VID:   VP4
  HwRev: V4.8
  New Deviation Number: 0
  CLEI:  IPUCARJBAA
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  MONLIB: QNXFFS Monlib Version 3.2
  ROMMON: Version 1.0(20081208:173612) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
    Compact Flash : V1.0
     XbarSwitch0  : V1.3
     XbarSwitch1  : V1.3
     XbarArbiter  : V1.0
     XbarInterface  : V0.0
     IntCtrl  : V1.14
     ClkCtrl  : V1.13
     PuntFPGA  : V1.5
     HD  : V3.0
     USB0  : V77.20
     USB1  : V77.20
     CPUCtrl  : V1.17
     UTI  : V1.6
     LIU  : V1.0
     MLANSwitch  : V0.0
     EOBCSwitch  : V2.0
     CBC (active partition)  : v1.2
     CBC (inactive partition)  : v1.1

NODE module 0/1/CPU0 :

  MAIN:  board type 0x20207
  S/N:   FOC123081J6
  Top Assy. Number:   68-3182-03
  PID:   A9K-40GE-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PortCtrl  : V0.8
     PHYCtrl  : V0.6
     40 Port Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
     CBC (inactive partition)  : v2.1

NODE module 0/4/CPU0 :

  MAIN:  board type 0x2020a
  S/N:   FOC123081JA
```

```
  Top Assy. Number:   68-3183-02
  PID:   A9K-8T/4-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  IPU3AE0CAA
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PortCtrl  : V0.10
     PHYCtrl  : V0.7
     PHY0  : V0.16
     PHY1  : V0.16
     PHY2  : V0.16
     PHY3  : V0.16
     PHY4  : V0.16
     PHY5  : V0.16
     PHY6  : V0.16
     PHY7  : V0.16
     8 Port Ten Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
     CBC (inactive partition)  : v2.1

NODE module 0/6/CPU0 :

  MAIN:  board type 0x20208
  S/N:   FHH12250033
  Top Assy. Number:   68-3184-02
  PID:   A9K-4T-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PHY0  : V0.16
     PHY1  : V0.16
     PHY2  : V0.16
     PHY3  : V0.16
     PortCtrl  : V0.10
     PHYCtrl  : V0.7
     4 Port Ten Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
```

```
     CBC (inactive partition)  : v2.1
```

In the following example, the **show diag** command displays information for a single node:

```
RP/0/RSP0/CPU0:router# show diag 0/6/cpu0

Mon Jun 29 00:41:43.450 PST

NODE module 0/6/CPU0 :

  MAIN:  board type 0x20208
  S/N:   FHH12250033
  Top Assy. Number:   68-3184-02
  PID:   A9K-4T-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PHY0  : V0.16
     PHY1  : V0.16
     PHY2  : V0.16
     PHY3  : V0.16
     PortCtrl  : V0.10
     PHYCtrl  : V0.7
     4 Port Ten Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
     CBC (inactive partition)  : v2.1
```

Displaying System Hardware Version Information

To display hardware version information for all or some of the components assigned in a system, connect to the designated shelf controller (DSC) and enter the **show diag** command in administration EXEC mode. When this command is entered in administration EXEC mode, you can display information on RSPs, line cards, and system components such as the chassis, fan trays, and power supplies.

Note If you enter the **show diag** command in EXEC mode, the software displays only the hardware assigned to the SDR to which you are connected.

The syntax for the **show diag** command in administration EXEC mode is:

show diag [*node-id* | **chassis** | **details** | **fans** | **memory** | **power-supply** | **summary**]

Tip For information on the software version, use the **show version** command.

In the following example, the **show diag** command displays information for all nodes in the system:

```
RP/0/RSP0/CPU0:router(admin)# show diag

Mon Jun 29 01:21:04.571 PST

NODE module 0/RSP0/CPU0 :

  MAIN:  board type 0x100302
  S/N:   FOC1230803H
  Top Assy. Number:   68-3160-04
  PID:   A2K-RSP-4G-HDD=
  UDI_VID:   VP4
  HwRev: V4.8
  New Deviation Number: 0
  CLEI:  IPUCARJBAA
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  MONLIB: QNXFFS Monlib Version 3.2
  ROMMON: Version 1.0(20081208:173612) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
    Compact Flash : V1.0
     XbarSwitch0  : V1.3
     XbarSwitch1  : V1.3
     XbarArbiter  : V1.0
     XbarInterface  : V0.0
     IntCtrl  : V1.14
     ClkCtrl  : V1.13
     PuntFPGA  : V1.5
     HD  : V3.0
     USB0  : V77.20
     USB1  : V77.20
     CPUCtrl  : V1.17
     UTI  : V1.6
     LIU  : V1.0
     MLANSwitch  : V0.0
     EOBCSwitch  : V2.0
     CBC (active partition)  : v1.2
     CBC (inactive partition)  : v1.1

NODE fantray 0/FT0/SP :

  MAIN:  board type 0x900211
  S/N:
  Top Assy. Number:   32-0000-00
  PID:
  UDI_VID:
  HwRev: V32.0
  New Deviation Number: 0
  CLEI:
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:
     CBC (active partition)  : v4.0
     CBC (inactive partition)  : v0.13
```

```
NODE fantray 0/FT1/SP :

  MAIN:  board type 0x900211
  S/N:
  Top Assy. Number:   32-0000-00
  PID:
  UDI_VID:
  HwRev: V32.0
  New Deviation Number: 0
  CLEI:
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:
     CBC (active partition)  : v4.0
     CBC (inactive partition)  : v0.13

NODE module 0/1/CPU0 :

  MAIN:  board type 0x20207
  S/N:   FOC123081J6
  Top Assy. Number:   68-3182-03
  PID:   A9K-40GE-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PortCtrl  : V0.8
     PHYCtrl  : V0.6
     40 Port Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
     CBC (inactive partition)  : v2.1

NODE module 0/4/CPU0 :

  MAIN:  board type 0x2020a
  S/N:   FOC123081JA
  Top Assy. Number:   68-3183-02
  PID:   A9K-8T/4-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  IPU3AE0CAA
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
```

```
      XbarInterface  : V18.4
      Bridge0  : V0.38
      Bridge1  : V0.38
      CPUCtrl  : V0.15
      USB  : V77.20
      PortCtrl  : V0.10
      PHYCtrl  : V0.7
      PHY0  : V0.16
      PHY1  : V0.16
      PHY2  : V0.16
      PHY3  : V0.16
      PHY4  : V0.16
      PHY5  : V0.16
      PHY6  : V0.16
      PHY7  : V0.16
      8 Port Ten Gigabit Ethernet Daughter board  : V0.0
      CBC (active partition)  : v2.2
      CBC (inactive partition)  : v2.1

NODE module 0/6/CPU0 :

  MAIN:  board type 0x20208
  S/N:   FHH12250033
  Top Assy. Number:   68-3184-02
  PID:   A9K-4T-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
      NP0  : V3.194
      NP1  : V3.194
      NP2  : V3.194
      NP3  : V3.194
      XbarInterface  : V18.4
      Bridge0  : V0.38
      Bridge1  : V0.38
      CPUCtrl  : V0.15
      USB  : V77.20
      PHY0  : V0.16
      PHY1  : V0.16
      PHY2  : V0.16
      PHY3  : V0.16
      PortCtrl  : V0.10
      PHYCtrl  : V0.7
      4 Port Ten Gigabit Ethernet Daughter board  : V0.0
      CBC (active partition)  : v2.2
      CBC (inactive partition)  : v2.1

NODE power-module 0/PM0/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
```

```
  Board FPGA/CPLD/ASIC Hardware Revision:

NODE power-module 0/PM1/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:

NODE power-module 0/PM2/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:

Rack 0 - ASR-9010 Chassis, Includes Accessories
  RACK NUM: 0
  S/N:
  PID:   ASR-9010 Backplane
  VID:   0.1
  Desc:  ASR-9010 Chassis, Includes Accessories
  CLEI:  NOCLEI
  Top Assy. Number:   68-1234-56
```

In the following example, the **show diag** command displays information for a single system component:

```
RP/0/RSP0/CPU0:router(admin)# show diag chassis

Mon Jun 29 01:25:05.711 PST

Rack 0 - ASR-9010 Chassis, Includes Accessories
  RACK NUM: 0
  S/N:
  PID:   ASR-9010 Backplane
  VID:   0.1
  Desc:  ASR-9010 Chassis, Includes Accessories
  CLEI:  NOCLEI
  Top Assy. Number:   68-1234-56
```

Displaying Software and Hardware Information

The **show version** command displays a variety of system information, including the hardware and software versions, router uptime, boot settings (including the configuration register), and active software.

The following is sample output from the **show version** command:

For Cisco ASR 9000 64 bit environment:

```
RP/0/RSP0/CPU0:Router#show version
Cisco IOS XR Software, Version 7.5.2
Copyright (c) 2013-2022 by Cisco Systems, Inc.

Build Information:
 Built By     : ingunawa
 Built On     : Tue Apr 26 18:26:36 PDT 2022
 Built Host   : iox-ucs-055
 Workspace    : /auto/srcarchive14/prod/7.5.2/asr9k-x64/ws
 Version      : 7.5.2
 Location     : /opt/cisco/XR/packages/
 Label        : 7.5.2

cisco ASR9K () processor
System uptime is 4 weeks 2 hours 31 minutes
```

For Cisco ASR9000 32 bit environment:

```
RP/0/RSP0/CPU0:Router#show version
Sat Aug  1 22:52:39.089 DST
Cisco IOS XR Software, Version 3.9.0.16I[DT_IMAGE]
Copyright (c) 2009 by Cisco Systems, Inc.
ROM: System Bootstrap, Version 1.1(20090521:183759) [ASR9K ROMMON],
router uptime is 1 day, 2 hours, 34 minutes
System image file is "bootflash:disk0/asr9k-os-mbi-3.9.0.16I/mbiasr9k-rp.vm"
cisco ASR9K Series (MPC8641D) processor with 4194304K bytes of memory.
MPC8641D processor at 1333MHz, Revision 2.2
2 Management Ethernet
12 TenGigE
40 GigabitEthernet
219k bytes of non-volatile configuration memory.
975M bytes of compact flash card.
33994M bytes of hard disk.
1605616k bytes of disk0: (Sector size 512 bytes).
1605616k bytes of disk1: (Sector size 512 bytes).
Configuration register on node 0/RSP0/CPU0 is 0x102
Boot device on node 0/RSP0/CPU0 is disk0:
Package active on node 0/RSP0/CPU0:
asr9k-scfclient, V 3.9.0.16I[DT_IMAGE], Cisco Systems, at disk0:asr9k-scfclient-3.9.0.16I
    Built on Thu Jul 30 12:09:40 DST 2009
    By sjc-lds-208 in /auto/ioxbuild7/production/3.9.0.16I.DT_IMAGE/asr9k/workspace for
c4.2.1-p0
asr9k-adv-video, V 3.9.0.16I[DT_IMAGE], Cisco Systems, at disk0:asr9k-adv-video-3.9.0.16I
    Built on Thu Jul 30 13:49:37 DST 2009
    By sjc-lds-208 in /auto/ioxbuild7/production/3.9.0.16I.DT_IMAGE/asr9k/workspace for
c4.2.1-p0
asr9k-fpd, V 3.9.0.16I[DT_IMAGE], Cisco Systems, at disk0:asr9k-fpd-3.9.0.16I
    Built on Thu Jul 30 12:26:21 DST 2009
    By sjc-lds-208 in /auto/ioxbuild7/production/3.9.0.16I.DT_IMAGE/asr9k/workspace for
c4.2.1-p0
asr9k-diags, V 3.9.0.16I[DT_IMAGE], Cisco Systems, at disk0:asr9k-diags-3.9.0.16I
    Built on Thu Jul 30 12:09:43 DST 2009
    By sjc-lds-208 in /auto/ioxbuild7/production/3.9.0.16I.DT_IMAGE/asr9k/workspace for
c4.2.1-p0
asr9k-k9sec, V 3.9.0.16I[DT_IMAGE], Cisco Systems, at disk0:asr9k-k9sec-3.9.0.16I
    Built on Thu Jul 30 12:25:25 DST 2009
    By sjc-lds-208 in /auto/ioxbuild7/production/3.9.0.16I.DT_IMAGE/asr9k/workspace for
c4.2.1-p0
asr9k-mgbl, V 3.9.0.16I[DT_IMAGE], Cisco Systems, at disk0:asr9k-mgbl-3.9.0.16I
```

```
   Built on Thu Jul 30 13:48:16 DST 2009
--More--
```

Displaying SDR Node IDs and Status

In EXEC mode, the **show platform** command displays information for all nodes assigned to the owner SDR. For each node, this information includes the host card type, the operational state, and the configuration state. To display information on a single node, enter the command with a node ID.

The syntax for the **show platform** command is:

show platform [*node-id*]

The following example displays the status for all nodes in the SDR to which you are connected:

```
RP/0/RSP0/CPU0:router# show platform
Mon Aug  3 07:39:01.416 DST
Node            Type                      State           Config State
-----------------------------------------------------------------------------
0/RSP0/CPU0     A9K-RSP-4G(Active)        IOS XR RUN      PWR,NSHUT,MON
0/1/CPU0        A9K-40GE-B                IOS XR RUN      PWR,NSHUT,MON
0/4/CPU0        A9K-8T/4-B                IOS XR RUN      PWR,NSHUT,MON
0/6/CPU0        A9K-4T-B                  IOS XR RUN      PWR,NSHUT,MON
```

The *node-id* appears in the *rack*/*slot*/*module* notation, and the *node-id* components are as follows:

- *rack* —In a single-shelf system the rack number is always “0.”
- *slot* —Number of the physical slot in which the card is installed.
- *module* —Subslot number of a system hardware component.

Table 10: Node ID Components, on page 113 summarizes the *node-id* for each type of card.

Table 10: Node ID Components

Card Type (the card to which your are issuing commands)	Rack (always “0”)	Slot (the physical slot in which the card is installed)	Module (the entity on the card that is the target of the command)
Route switch processor	0	RSP0 and RSP1	CPU0
40-Port Gigabit Ethernet Line Card 8-Port 10-Gigabit Ethernet Line Card 4-Port 10-Gigabit Ethernet Line Card	0-255	4-7 (6-slot chassis) 0–7 (10-slot chassis)	0-X (SFP and XFP module number on the line card)
Power Modules	0	PM0-PM5 (10-slot chassis) PM0-PM2 (6-slot chassis	—
Fan controller cards	0	FC0–FC1	—

Displaying Router Node IDs and Status

In administration EXEC mode, the **show platform** command displays information for all router nodes. In administration EXEC mode, the command display also includes additional node IDs such as those for fabric cards, alarm modules, and fan controllers. For each node, this information includes the host card type, the operational state, and the configuration state. To display information on a single node, enter the command with a node ID.

The syntax for the **show platform** command is:

show platform [*node-id*]

The following example displays the status for all nodes in the system:

```
RP/0/RSP0/CPU0:router(admin)# show platform

Sat Mar 24 05:02:18.569 DST
Node            Type                      State            Config State
-----------------------------------------------------------------------------
0/RSP0/CPU0     A9K-RSP-4G(Active)        IOS XR RUN       PWR,NSHUT,MON
0/1/CPU0        A9K-40GE-B                IOS XR RUN       PWR,NSHUT,MON
0/4/CPU0        A9K-8T/4-B                IOS XR RUN       PWR,NSHUT,MON
0/6/CPU0        A9K-4T-B                  IOS XR RUN       PWR,NSHUT,MON
```

The *node-id* appears in the *rack*/*slot*/*module* notation, and the *node-id* components are as follows:

- *rack* —In a single-shelf system the rack number is always “0.”
- *slot* —Number of the physical slot in which the card is installed.
- *module* —Subslot number of a system hardware component.

Table 10: Node ID Components, on page 113 summarizes the *node-id* argument for each type of card.

Displaying Router Environment Information

The **show environment** command displays hardware information for the system, including fan speeds, LED indications, power supply voltage and current information, and temperatures.

The syntax for the **show environment** command is:

show environment [*options*]

You can use the **show environment** command options to limit the detail in the command display. To view the command options, enter the **show environment ?** command. The following example shows the full environment status report:

```
RP/0/RSP0/CPU0:router(admin)# show environment

Mon Jun 29 04:32:07.587 PST

Temperature Information
---------------------------------------------

R/S/I   Modules         Inlet           Hotspot
                        Temperature     Temperature
```

```
                       (deg C)         (deg C)


0/1/*
        host             31.5           39.5

0/RSP0/*
        host             26.6           36.6

0/4/*
        host             29.8           38.8

0/6/*
        host             32.7           42.0

0/FT0/*
        host             27.2           28.2
0/FT1/*
        host             27.4           30.2

Voltage Information
---------------------------------------------

R/S/I   Modules Sensor                  (mV)   Margin

0/1/*
        host    IBV                    10647   n/a
        host    5.0V                    4929   n/a
        host    VP3P3_CAN               3288   n/a
        host    3.3V                    3301   n/a
        host    2.5V                    2516   n/a
        host    1.8VB                   1810   n/a
        host    1.2VB                   1193   n/a
        host    1.8VA                   1800   n/a
        host    0.9VB                    884   n/a
        host    1.2V_LDO_BRG0           1193   n/a
        host    1.2V_LDO_BRG1           1195   n/a
        host    1.8VC                   1811   n/a
        host    1.5VB                   1505   n/a
        host    1.5VA                   1503   n/a
        host    1.1V(1.05V_CPU)         1052   n/a
        host    0.75VA                   751   n/a
        host    0.75VB_0.75VC            754   n/a
        host    1.1VB                   1102   n/a
        host    1.2V_TCAM0              1003   n/a
        host    1.2V_TCAM1              1000   n/a
        host    1.0V_Bridge_LDO          998   n/a
        host    1.0VB                   1043   n/a
        host    0.75VD_and_0.75VE        752   n/a
        host    1.2V_TCAM2              1005   n/a
        host    1.2V_TCAM3              1002   n/a
        host    1.5VC                   1504   n/a
        host    1.8VD                   1803   n/a
        host    1.1VC                   1099   n/a
        host    ZARLINK_3.3V            3272   n/a
        host    ZARLINK_1.8V            1808   n/a
        host    1.2V_DB                 1195   n/a
        host    3.3V_DB                 3316   n/a
        host    2.5V_DB                 2534   n/a
        host    1.5V_DB                 1509   n/a

0/RSP0/*
        host    0.75VTT                  749   n/a
        host    0.9VTT_A                 910   n/a
```

```
          host    0.9VTT_B                 904   n/a
          host    IBV                    10586   n/a
          host    5.0V                    5013   n/a
          host    VP3P3_CAN               3277   n/a
          host    3.3V                    3299   n/a
          host    2.5V                    2518   n/a
          host    1.8VB                   1807   n/a
          host    1.2VA                   1205   n/a
          host    1.2VB                   1202   n/a
          host    1.05V                   1047   n/a
          host    1.2VD                   1205   n/a
          host    1.8VA                   1811   n/a
          host    1.5V                    1496   n/a
          host    1.9V                    1887   n/a

0/4/*
          host    IBV                    10627   n/a
          host    5.0V                    4917   n/a
          host    VP3P3_CAN               3279   n/a
          host    3.3V                    3296   n/a
          host    2.5V                    2522   n/a
          host    1.8VB                   1805   n/a
          host    1.2VB                   1188   n/a
          host    1.8VA                   1796   n/a
          host    0.9VB                    881   n/a
          host    1.2V_LDO_BRG0           1192   n/a
          host    1.2V_LDO_BRG1           1195   n/a
          host    1.8VC                   1806   n/a
          host    1.5VB                   1510   n/a
          host    1.5VA                   1503   n/a
          host    1.1V(1.05V_CPU)         1048   n/a
          host    0.75VA                   753   n/a
          host    0.75VB_0.75VC            757   n/a
          host    1.1VB                   1105   n/a
          host    1.2V_TCAM0              1003   n/a
          host    1.2V_TCAM1              1000   n/a
          host    1.0V_Bridge_LDO          997   n/a
          host    1.0VB                   1037   n/a
          host    0.75VD_and_0.75VE        755   n/a
          host    1.2V_TCAM2              1004   n/a
          host    1.2V_TCAM3              1005   n/a
          host    1.5VC                   1505   n/a
          host    1.8VD                   1808   n/a
          host    1.1VC                   1104   n/a
          host    ZARLINK_3.3V            3285   n/a
          host    ZARLINK_1.8V            1806   n/a
          host    1.2V_DB                 1205   n/a
          host    3.3V_DB                 3318   n/a
          host    2.5V_DB                 2493   n/a
          host    1.5V_DB                 1497   n/a
          host    1.8V_DB                 1825   n/a
          host    5.0V_XFP_DB             5001   n/a
          host    1.2VB_DB                1228   n/a

0/6/*
          host    IBV                    10628   n/a
          host    5.0V                    4893   n/a
          host    VP3P3_CAN               3281   n/a
          host    3.3V                    3297   n/a
          host    2.5V                    2524   n/a
          host    1.8VB                   1804   n/a
          host    1.2VB                   1204   n/a
          host    1.8VA                   1795   n/a
          host    0.9VB                    881   n/a
```

```
        host    1.2V_LDO_BRG0            1194   n/a
        host    1.2V_LDO_BRG1            1193   n/a
        host    1.8VC                    1815   n/a
        host    1.5VB                    1495   n/a
        host    1.5VA                    1503   n/a
        host    1.1V(1.05V_CPU)          1052   n/a
        host    0.75VA                    752   n/a
        host    0.75VB_0.75VC             749   n/a
        host    1.1VB                    1001   n/a
        host    1.2V_TCAM0                999   n/a
        host    1.2V_TCAM1               1002   n/a
        host    1.0V_Bridge_LDO           995   n/a
        host    1.0VB                    1050   n/a
        host    0.75VD_and_0.75VE         752   n/a
        host    1.2V_TCAM2               1002   n/a
        host    1.2V_TCAM3                995   n/a
        host    1.5VC                    1502   n/a
        host    1.8VD                    1802   n/a
        host    1.1VC                    1101   n/a
        host    ZARLINK_3.3V             3273   n/a
        host    ZARLINK_1.8V             1804   n/a
        host    1.2V_DB                  1200   n/a
        host    3.3V_DB                  3314   n/a
        host    2.5V_DB                  2496   n/a
        host    1.5V_DB                  1496   n/a
        host    1.8V_DB                  1824   n/a
        host    5.0V_XFP_DB              5004   n/a
        host    1.2VB_DB                 1227   n/a

LED Information
---------------------------------------------

R/S/I   Modules LED             Status

0/RSP0/*
        host    Critical-Alarm  Off
        host    Major-Alarm     Off
        host    Minor-Alarm     Off
        host    ACO             Off

Fan Information
---------------------------------------------

  Fan speed (rpm):
        FAN0    FAN1    FAN2    FAN3    FAN4    FAN5    FAN6    FAN7    FAN8    FAN9
FAN10   FAN11

0/FT0/*
        3510    3510    3510    3540    3510    3570    3480    3570    3510    3510
3510    3510
0/FT1/*
        3540    3510    3450    3540    3480    3600    3480    3450    3540    3540
3480    3540

Power Supply Information
---------------------------------------------

R/S/I   Modules Sensor          Watts

0/PM0/*
        host    PM              3000

0/PM1/*
        host    PM              3000
```

```
0/PM2/*
        host    PM              3000

Power Shelves Type: AC

Total Power Capacity:                  9000W
Protected Power Capacity:              4500W
Worst Case Power Used:                 3145W
 Slot                                                     Max Watts
 ----                                                     ---------
 0/1/CPU0                                                       375
 0/RSP0/CPU0                                                    250
 0/RSP1/CPU0                                                    350
 0/4/CPU0                                                       375
 0/6/CPU0                                                       375
 0/FT0/SP                                                       710  (default)
 0/FT1/SP                                                       710  (default)

Worst Case Protected Power Available:   1355W
```

Configuring the Chassis Altitude

To allow your router to adjust the fan speed to compensate for lower cooling capabilities at higher altitudes, you should configure the chassis altitude setting. Use the **environment altitude** command in administration configuration mode. The default setting is 1800 meters.

The syntax for the environment altitude command is:

environment altitude *altitude* **rack** *rack-no*

Displaying RP Redundancy Status

The **show redundancy** command displays the redundancy status of the route switch processors (RSPs). This command also displays the boot and switch-over history for the RSPs.

The **show redundancy** operates in EXEC and administration EXEC mode.

In the following example, the **show redundancy** command displays the redundancy status for a redundant RSP pair:

```
RP/0/RSP0/CPU0:router(admin)# show redundancy

Mon Jun 29 04:49:26.098 PST
Redundancy information for node 0/RSP0/CPU0:
==========================================
Node 0/RSP0/CPU0 is in ACTIVE role
Node 0/RSP0/CPU0 has no valid partner

Reload and boot info
----------------------
A9K-RSP-4G reloaded Thu Jun 11 15:20:50 2009: 2 weeks, 3 days, 13 hours, 28 minutes ago
Active node booted Thu Jun 11 15:20:50 2009: 2 weeks, 3 days, 13 hours, 28 minutes ago

Active node reload "Cause: Turboboot completed successfully"
```

Displaying Field-Programmable Device Compatibility

The **show hw-module fpd** command displays field-programmable device (FPD) compatibility for all modules or a specific module.

The syntax for the **show hw-module fpd** command is:

show hw-module fpd location {**all** | *node-id*}

The **show hw-module fpd** operates in EXEC and administration EXEC mode.

The following example shows how to display FPD compatibility for all modules in the router:

```
RP/0/RSP0/CPU0:router# ios#show hw-module fpd
Tue Jan 22 13:56:55.082 UTC
                                                               FPD Versions
                                                               =================
Location   Card type        HWver FPD device       ATR Status   Running Programd
-------------------------------------------------------------------------------
0/RP0     NCS-55A2-MOD-S    0.3   MB-MIFPGA            CURRENT    0.19    0.19
0/RP0     NCS-55A2-MOD-S    0.3   Bootloader           CURRENT    1.10    1.10
0/RP0     NCS-55A2-MOD-S    0.3   CPU-IOFPGA           CURRENT    1.18    1.18
0/RP0     NCS-55A2-MOD-S    0.3   MB-IOFPGA            CURRENT    0.18    0.18
0/PM0     NC55-1200W-ACFW   1.0   LIT-PriMCU-ACFW      NEED UPGD  2.08    2.08
0/PM1     NC55-1200W-ACFW   1.0   LIT-PriMCU-ACFW      NEED UPGD  2.08    2.08
RP/0/RP0/CPU0:ios#.
```

Note After Release 5.3.x, Upg/Dng? will display Yes only for upgrade.

The following example shows the FPD for which upgrage will be skipped.

```
RP/0/RP0/CPU0:router# show hw-module fpd location all

===================================== ==========================================
                                      Existing Field Programmable Devices
                                      ==========================================
                                        HW                       Current SW Upg/
Location     Card Type                Version Type Subtype Inst   Version   Dng?
============ ======================== ======= ==== ======= ==== =========== ====
0/SM1/SP     140G-4-S1S2S3              0.1  lc   rommonA   0    2.08       Yes
--------------------------------------------------------------------------------
                                             lc    rommon   0    2.08       Yes
--------------------------------------------------------------------------------
                                             lc    fpga1    0    6.04^      No
--------------------------------------------------------------------------------
                                             lc    fpga2    0    4.01       No
--------------------------------------------------------------------------------
NOTES:
1. ^ One or more FPD will be intentionally skipped from upgrade using CLI with option "all"
 or during "Auto fpd".
   It can be upgraded only using the "admin> upgrade hw-module fpd <fpd> location <loc>"
CLI with exact location.

RP/0/RSP1/CPU0:router# show hw-module fpd location all

Mon Jun 29 05:38:50.332 PST
```

```
===================================== ==========================================
                                      Existing Field Programmable Devices
                                      ==========================================
                                        HW                       Current SW Upg/
Location     Card Type                Version Type Subtype Inst   Version   Dng?
============ ======================== ======= ==== ======= ==== =========== ====
0/RSP0/CPU0  A9K-RSP-4G                 4.8   lc   fpga3   0       1.13     No
                                              lc   fpga1   0       1.5      No
                                              lc   fpga2   0       1.14     No
                                              lc   cbc     0       1.2      No
                                              lc   fpga4   0       1.6      No
                                              lc   rommon  0       1.0      No
--------------------------------------------------------------------------------
0/RSP0/CPU0  ASR-9010-FAN               1.0   lc   cbc     1       4.0      No
--------------------------------------------------------------------------------
0/RSP0/CPU0  ASR-9010-FAN               1.0   lc   cbc     2       4.0      No
--------------------------------------------------------------------------------
0/1/CPU0     A9K-40GE-B                 1.0   lc   fpga1   0       0.38     No
                                              lc   fpga2   0       0.8      No
                                              lc   cbc     0       2.2      No
                                              lc   cpld1   0       0.15     No
                                              lc   rommon  0       1.0      No
--------------------------------------------------------------------------------
0/1/CPU0     A9K-40GE-B                 1.0   lc   fpga1   1       0.38     No
--------------------------------------------------------------------------------
0/4/CPU0     A9K-8T/4-B                 1.0   lc   fpga1   0       0.38     No
                                              lc   fpga2   0       0.10     No
                                              lc   cbc     0       2.2      No
                                              lc   cpld2   0       0.7      No
                                              lc   cpld1   0       0.15     No
                                              lc   cpld3   0       0.3      No
                                              lc   rommon  0       1.0      No
                                              lc   fpga3   0      14.42     No
--------------------------------------------------------------------------------
0/4/CPU0     A9K-8T/4-B                 1.0   lc   fpga1   1       0.38     No
--------------------------------------------------------------------------------
0/6/CPU0     A9K-4T-B                   1.0   lc   fpga1   0       0.38     No
                                              lc   fpga2   0       0.10     No
                                              lc   cbc     0       2.2      No
                                              lc   cpld2   0       0.7      No
                                              lc   cpld1   0       0.15     No
                                              lc   cpld3   0       0.3      No
                                              lc   rommon  0       1.0      No
                                              lc   fpga3   0      14.42     No
--------------------------------------------------------------------------------
0/6/CPU0     A9K-4T-B                   1.0   lc   fpga1   1       0.38     No
--------------------------------------------------------------------------------
```

The following example shows how to display FPD compatibility for a specific module in the router:

Table 11: show hw-module fpd Field Descriptions

Field	Description
Location	Location of the module in the *rack*/*slot*/*module* notation.
Card Type	Module part number.
HW Version	Hardware model version for the module.

Field	Description
Type	Hardware type. Can be one of the following types: • spa—Shared port adapter • lc—Line card
Subtype	FPD type. Can be one of the following types: • fabldr—Fabric downloader • fpga1—Field-programmable gate array • fpga2—Field-programmable gate array 2 • fpga3—Field-programmable gate array 3 • fpga4—Field-programmable gate array 4 • fpga5—Field-programmable gate array 5 • rommonA—Read-only memory monitor A • rommon—Read-only memory monitor B
Inst	FPD instance. The FPD instance uniquely identifies an FPD and is used by the FPD process to register an FPD.
Current SW Version	Currently running FPD image version.
Upg/Dng?	Specifies whether an FPD upgrade or downgrade is required. A downgrade is required in rare cases when the version of the FPD image has a higher major revision than the version of the FPD image in the current Cisco IOS XR software package.

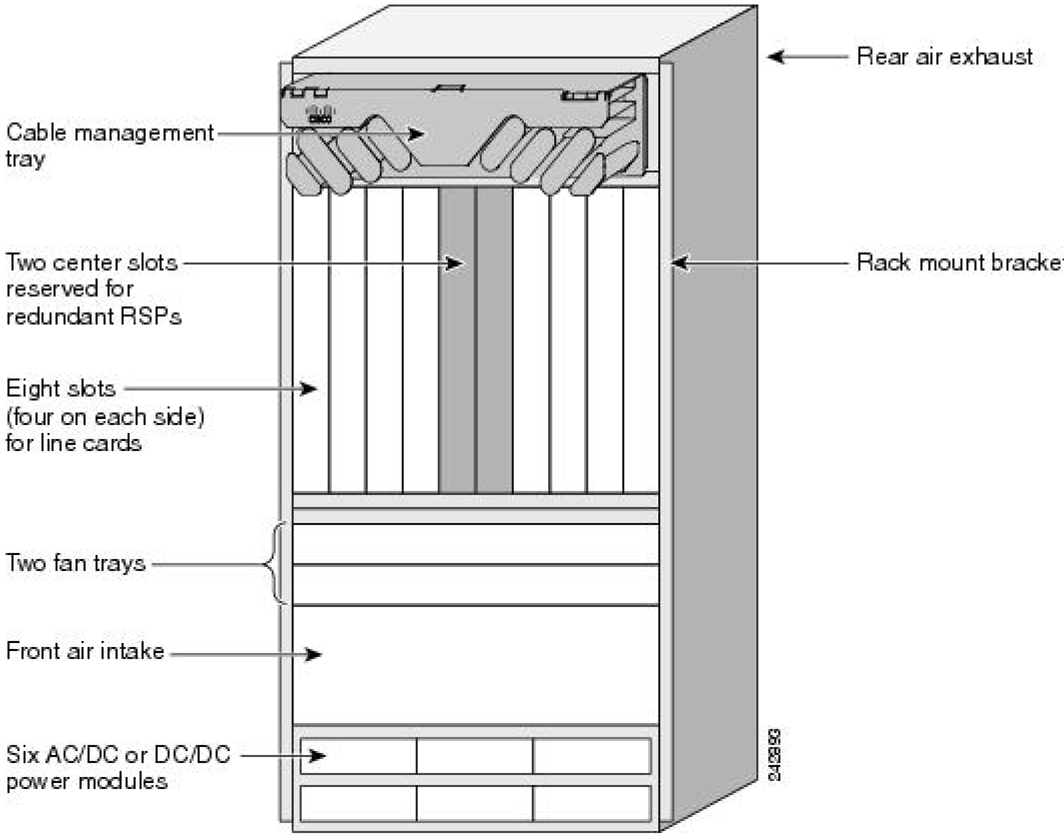
RSP Redundancy and Switchover

This section describes RSP redundancy and switchover commands and issues.

Establishing RSP Redundancy

Your router has two slots for RSPs: RSP0 and RSP1 (see Figure 7: Redundant Set of RSPs Installed in Slots RSP0 and RSP1 in an 8-Slot Chassis, on page 122). RSP0 is the slot on the left, facing the front of the chassis, and RSP1 is the slot on right. These slots are configured for redundancy by default, and the redundancy cannot be eliminated. To establish RSP redundancy, install RSPs into both slots.

Figure 7: Redundant Set of RSPs Installed in Slots RSP0 and RSP1 in an 8-Slot Chassis



Determining the Active RP in a Redundant Pair

During system startup, one RSP in each redundant pair becomes the active RSP. You can tell which RSP is the active RSP in the following ways:

- The active RSP can be identified by the green Primary LED on the faceplate of the card. The active RSP is indicated when the Primary LED is on. The alphanumeric LED display on the RSP displays ACTV RP.
- The slot of the active RSP is indicated in the CLI prompt. For example:

  ```
  RP/0/RSP1/CPU0:router#
  ```

  In this example, the prompt indicates that you are communicating with the active RSP in slot RSP1. See *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide* for a complete description of the CLI prompt.

- Enter the **show redundancy** command in EXEC mode to display a summary of the active and standby RSP status. For example:

  ```
  RP/0/RSP0/CPU0:router(admin)# show redundancy

  Mon Jun 29 04:49:26.098 PST
  Redundancy information for node 0/RSP0/CPU0:
  ==========================================
  ```

```
Node 0/RSP0/CPU0 is in ACTIVE role
Node 0/RSP0/CPU0 has no valid partner

Reload and boot info
----------------------
A9K-RSP-4G reloaded Thu Jun 11 15:20:50 2009: 2 weeks, 3 days, 13 hours, 28 minutes ago
Active node booted Thu Jun 11 15:20:50 2009: 2 weeks, 3 days, 13 hours, 28 minutes ago

Active node reload "Cause: Turboboot completed successfully"
```

Role of the Standby RSP

The second RSP to boot in a redundant pair automatically becomes the standby RSP. While the active RSP manages the system and communicates with the user interface, the standby RSP maintains a complete backup of the software and configurations for all cards in the system. If the active RSP fails or goes off line for any reason, the standby RSP immediately takes control of the system.

Summary of Redundancy Commands

RSP redundancy is enabled by default in the Cisco IOS XR software, but you can use the commands described in Table 12: RSP Redundancy Commands, on page 123 to display the redundancy status of the cards or force a manual switchover.

Table 12: RSP Redundancy Commands

Command	Description
show redundancy	Displays the redundancy status of the RSPs. This command also displays the boot and switch-over history for the RSPs.
redundancy switchover	Forces a manual switchover to the standby RSP. This command works only if the standby RSP is installed and in the "ready" state.
show platform	Displays the status for node, including the redundancy status of the RSP cards. In EXEC mode, this command displays status for the nodes assigned to the SDR. In administration EXEC mode, this command displays status for all nodes in the system.

Automatic Switchover

Automatic switchover from the active RSP to the standby RSP occurs only if the active RSP encounters a serious system error, such as the loss of a mandatory process or a hardware failure. When an automatic switchover occurs, the RSPs respond as follows:

- If a standby RSP is installed and "ready" for switchover, the standby RSP becomes the active RSP. The original active RSP attempts to reboot.
- If the standby RSP is not in "ready" state, then both RSPs reboot. The first RSP to boot successfully assumes the role of active RSP.

RSP Redundancy During RSP Reload

The **reload** command causes the active RSP to reload the Cisco IOS XR software. When an RSP reload occurs, the RSPs respond as follows:

- If a standby RSP is installed and “ready” for switchover, the standby RSP becomes the active RSP. The original active RSP reboots and becomes the standby RSP.
- If the standby RSP is not in the “ready” state, then both RSPs reboot. The first RSP to boot successfully assumes the role of active RSP.

> **Caution** You should not use the **reload** command to force an RSP switchover because the result could be a significant loss of router operations. Instead, use the **redundancy switchover** command to fail over to the standby RSP, then use the **hw-module location** *node-id* **reload** command to reload the new standby RSP.

Related Topics

Reloading, Shutting Down, or Power Cycling a Node, on page 138

Manual Switchover

You can force a manual switchover from the active RSP to the standby RSP using the **redundancy switchover** command.

If a standby RSP is installed and ready for switchover, the standby RSP becomes the active RSP. The original active RSP becomes the standby RSP. In the following example, partial output for a successful redundancy switchover operation is shown:

```
RP/0/RSP0/CPU0:router# show redundancy

  This node (0/RSP0/CPU0) is in ACTIVE role
  Partner node (0/RSP1/CPU0) is in STANDBY role
  Standby node in 0/RSP1/CPU0 is ready

  RP/0/RSP0/CPU0:router# redundancy switchover
  Updating Commit Database.  Please wait...[OK]
  Proceed with switchover 0/RSP0/CPU0 -> 0/RSP1/CPU0? [confirm]
  Initiating switch-over.
  RP/0/RSP0/CPU0:router#

  <Your 'TELNET' connection has terminated>
```

In the preceding example, the Telnet connection is lost when the previously active RP resets. To continue management of the router, you must connect to the newly activated RP as shown in the following example:

```
User Access Verification

  Username: xxxxx
  Password: xxxxx
  Last switch-over Sat Apr 15 12:26:47 2009: 1 minute ago
```

```
RP/0/RSP1/CPU0:router#
```

If the standby RSP is not in “ready” state, the switchover operation is not allowed. In the following example, partial output for a failed redundancy switchover attempt is shown:

```
RP/0/RSP0/CPU0:router# show redundancy

  Redundancy information for node 0/RP1/CPU0:
  ==========================================
  Node 0/RSP0/CPU0 is in ACTIVE role
  Partner node (0/RSP1/CPU0) is in UNKNOWN role

  Reload and boot info
  ----------------------
  RP reloaded Wed Mar 29 17:22:08 2009: 2 weeks, 2 days, 19 hours, 14 minutes ago
  Active node booted Sat Apr 15 12:27:58 2009: 8 minutes ago
  Last switch-over Sat Apr 15 12:35:42 2009: 1 minute ago
  There have been 4 switch-overs since reload

  RP/0/RSP0/CPU0:router# redundancy switchover

  Switchover disallowed: Standby node is not ready.
```

System Logs during RSP Switchover

Table 13: Feature History Table

Feature Name	Release Information	Feature Description
RSP Slot Location in Syslog	Release 7.8.1	When an RSP switchover occurs, the router logs the active RSP slot location in the syslog message. This helps you quickly identify the active RSP slot from your router's system log messages. In earlier releases, the RSP switchover syslog message didn't include the active RSP slot location.

In the event of an RSP switchover, the router logs the following syslog messages:

```
RP/0/1/CPU0:Feb 19 09:08:00.655 UTC: rmf_svr[436]: %HA-REDCON-6-GO_ACTIVE : this card going
 active
RP/1/1/CPU0:Mar 8 11:43:29.041 UTC: rmf_svr[147]: %HA-REDCON-6-GO_STANDBY : this card going
 standby, location RP/1/1/CPU0
```

From Cisco IOS XR Release 7.8.1 onwards, the RSP switchover syslog message for the active RSP includes the RSP slot location as well:

```
RP/0/1/CPU0:Mar  8 11:42:50.876 UTC: rmf_svr[165]: %HA-REDCON-6-GO_ACTIVE : this card going
 active , location RP/0/1/CPU0:
```

Communicating with a Standby RP

The active RSP automatically synchronizes all system software, settings, and configurations with the standby RSP.

If you connect to the standby RSP through the console port, you can view the status messages for the standby RSP. The standby RSP does not display a CLI prompt, so you cannot manage the standby card while it is in standby mode.

If you connect to the standby RSP through the management Ethernet port, the prompt that appears is for the active RSP, and you can manage the router the same as if you had connected through the management Ethernet port on the active RSP.

PFM Alarm on OCXO Failure

Table 14: Feature History Table

Feature Name	Release Information	Feature Description
PFM Alarm on OCXO Failure	Release 6.7.35	You now get notified via Platform Fault Manager (PFM) alarms when the internal oscillator Oven-Controlled Crystal Oscillator (OCXO) fails. OCXO failure causes timing loss in internal devices leading to possible interface flaps. This feature covers two scenarios: • In the case of a single RSP, you only receive a notification via the PFM alarm. • In the case of dual RSPs, RSP switchover takes place along with the notification. Use the show reboot command to view the Onboard Failure Logging (OBFL) logs and track the PFM alarms.

The function of the OCXO, which is an internal Oscillator, is to provide clock to the Phased-Locked Loop (PLL). OCXO is connected to PLL which is further connected to many hardware components. When OCXO fails to deliver the clock output, timing loss occurs. This may lead to further problems in the system such as interface flaps.

When the OCXO failure is detected, the PFM alarms are displayed on the router console. Contact the Cisco Support team when you see PFM alarms on router console as this is a hardware failure and you must get your hardware checked.

Note When OCXO fails, local oscillator clock output is unstable and the line card interfaces state might be unpredictable.

PFM Alarms

The following are the examples of the PFM alarms that are generated on the router console:

- Dual RSP

```
Reboot Reason: Cause code 0x2c00001b Cause: Card reset requested by: Process ID: 102448
 (dao_tmp), Target node: 0/RSP0/CPU0, CondID: 637, Fault Reason: RSP PLL error log:
Local oscillator clock output is unstable, Reloading the board. Process: pfm_node_rp
```

- Standalone RSP

```
RP/0/RSP1/CPU0:Mar 15 11:22:13.647 UTC: pfm_node_rp[365]:
%PLATFORM-CLKCTRL_T-3-RSP_OCXO_FAILURE_ALARM : Set|dao_tmp[102448]|0x1039000|RSP PLL
error log: Local oscillator clock output is unstable. Line card interfaces may be
unpredictable
```

Restrictions for PFM Alarms on OCXO Failure

PFM alarm generation during OCXO failure is supported on the following Route Processor (RP)/Route Switch Processor (RSP) only:

- A9K-RSP880-LT-SE/TR
- A99-RP2-SE/TR
- A9K-RSP880-SE/TR

Mitigation

The router tries to recover from OCXO failure by performing switchover of RSPs in the case of Dual RSPs.

Use the **show platform** command to detect whether your router has standalone RSP or dual RSPs.

Here, is an example of command output for dual RSP:

```
Router#show platform
Node              Type                        State            Config state
--------------------------------------------------------------------------------
0/RSP0/CPU0       A9K-RSP880-SE(Active)       IOS XR RUN       NSHUT
0/RSP1/CPU0       A9K-RSP880-SE(Standby)      IOS XR RUN       NSHUT
0/FT0             ASR-9904-FAN                OPERATIONAL      NSHUT
0/0               A9K-48X10GE-1G-TR           SW_INACTIVE      NSHUT
0/1/CPU0          A9K-4X100GE-SE              IOS XR RUN       NSHUT
0/PT0             A9K-AC-PEM-V2               OPERATIONAL      NSHUT
Router#
```

Here, is an example of command output for standalone RSP:

```
Router#show platform
Node              Type                        State            Config state
--------------------------------------------------------------------------------
0/RSP0/CPU0      A9K-RSP880-SE (Active)          IOS XR RUN         NSHUT
0/FT0             ASR-9901-FAN                OPERATIONAL      NSHUT
0/FT1             ASR-9901-FAN                OPERATIONAL      NSHUT
```

```
0/FT2             ASR-9901-FAN               OPERATIONAL       NSHUT
0/0/CPU0          ASR9901-LC                 IOS XR RUN        NSHUT
0/PT0             A9K-AC-PEM                 OPERATIONAL       NSHUT
Router#
```

Dual RSPs

When the router has High Availabilty (HA) RSPs or Dual RSPs, there are two RSPs: RSP0 and RSP1. One of them is active RSP and other is standby RSP. When OCXO fails, you can see a PFM alarm on the router console. The router performs RSP switchover to recover from the OCXO failure and to maintain the timing accuracy. During switchover, the active RSP becomes the standby RSP and the standby RSP takes the role of active RSP. RSP switchover takes maximum 10 seconds.

However, you must contact Cisco Support team.

To track the fatal PFM alarms generated, the router maintains OBFL logs. .

Use the **Router# show reboot history location 0/RSP0/CPU0** command to view the OBFL logs.

Here, is a sample output of reboot history.

```
No Time Cause code reason
01 Thu Nov 11 13:54:54 2021  0x2c00001b Cause: Card reset requested by: Process ID: 102448
 (dao_tmp), Target node: 0/RSP0/CPU0, CondID: 637, Fault Reason: RSP PLL error log: Local
oscillator clock output is unstable, Reloading the board.
Process: pfm_node_rp

02 Fri Nov 12 04:26:11 2021  0x0400004f Cause: MBI-HELLO reloading node on receiving reload
 notification
Process: mbi-hello
```

Standalone RSP

When only one RSP is in use, you can only see the PFM alarm on the router console. This alarm indicates that OCXO failure has occurred and you must contact Cisco Support team.

Console Management Port

The Console Management Port (CMP) feature enables console access to the RSP and RP network devices through an ethernet port on the router using the Secure Shell (SSH).

To enable CMP feature the IPU and ROMMON must be upgraded to the latest version available in the Cisco IOS XR Software Release 5.3.2 through FPD upgrade for IOS XR 32-bit image, and Cisco IOS XR Software Release 6.4.1 for IOS XR 64-bit image. .

For information about FPD upgrade, see *Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide*, chapter *Upgrading FPD*.

Note

- CMP feature helps troubleshoot the RP and RSP issues when IOS XR CLI is unavailable or when the CPU is inaccessible. On the contrary, using the CMP feature otherwise will result in unpredictable behavior of the router.
- CMP is supported only on RSP 880, RSP880-LT, RSP5, RP2, and RP3 hardware.

The CMP feature enables:

- Connection to route processor console port.
- Connection to route processor auxiliary port (32-bit image) or system admin plane (64-bit image).
- installation of new software image through SCP (32-bit image) or PXE (64-bit image) without a terminal server connected to the console port.
- CMP password recovery by using the **resetcmp** command on the CMP shell. This clears CMP data (user IDs, passwords, DNS name, hostname, SSH Key) to default settings.

Note The default login username is cmp and password is cisco.

You can download a new IOS XR 32-bit image using the **scpboot** command (image will be turbo booted), and a new IOS XR 64-bit image using the **pxeboot** command.. You must provide the server IPv6 address and filename when using **scpboot** command. The image is copied from the server directly to the route processor CPU memory. If route processor CPU side is in ROMMON or already in IOS XR, it is reset and held in ROMMON until the image is copied. This image is automatically booted (turbo boot for 32-bit and pxeboot for 64-bit image) on the route processor CPU side. The image download options (scpboot and pxeboot) provided by the CMP can only download and boot a complete image. Subsequent image upgrades, pie downloads (32-bit image) and VM downloads (64-bit image) must be done through system admin (32-bit image), XR (64-bit image) and using the management ports.

CMP implements zero-configuration networking concepts such as mDNS and DNS-SD to ease the booting of a supervisor (RSP, RP) card. See the section Zero Configuration Networking, on page 131 for information on zero-configuration networking.

For information on CMP shell, see the section CMP Shell, on page 129.

CMP Shell

CMP is accessed using IPv6 SSH. Use the default username/password to login to CMP shell. This table describes the commands available on the CMP shell:

Table 15: CMP Shell Commands

Command	Description
adduser	Adds a new CMP user ID/password.
aux	Connects to route processor CPU auxiliary port for 32-bit image. Connects to system admin plane for 64-bit image.
con	Connects to route processor CPU console port. Although multiple SSH sessions to the CMP shell are allowed, the **con** , **aux**, or **lc** command execution is allowed for only single user at a time.
copykey	SCP a key.

Command	Description
deluser	Deletes a user ID. It is recommended that you delete the default username cmp after a new user is created.
desc_err	Shows description of command error codes.
debug	Enables CMP console logging functionality.
dns	Changes DNS name. The initial service advertisement uses the domain name of chassis serial number + RSP/RP slot. This can be changed using the **dns** command.
exit	Logs out of CMP.
fanspeed	Shows information about fan trays in the chassis.
help	Displays available CMP commands.
hostname	Changes a host name.
lc_con	Connects to a line card console.
lcslotinfo	Shows line card slot ID information.
passwd	Changes password (minimum 5 and maximum 8 characters).
priv	Enters privileged EXEC mode.
pxeboot	PXE boots a 64-bit Cisco IOS XR image to Route processor CPU memory.
resetcmp	Clears CMP data (user IDs, passwords, DNS name, hostname and SSH key) to default settings.
run	Runs diagnostic commands ping/ping6/traceroute/traceroute6 to diagnose basic network connectivity problems.
scpboot	SCP boots 32-bit IOS XR image and TURBOBOOT to route processor CPU memory.
show	Shows all CMP data. Displays ip/key/cmp configuration.
showinv	Shows the physical inventory.
showtemp	Shows the temperature information.
slotmap	Displays physical slot and card mapping information.

Command	Description
sshkeygen	Generates a new SSH key.
unlock	Removes all system locks. From CMP shell only one user is allowed to login to the console port, auxiliary port or LC console, and that user holds the lock and there is no access to other users.
warmreset	Warm resets local route processor.

Return Material Authorization (RMA) - In the event of a RMA of the supervisor (RSP/RP) card, since the CMP information is tied to the chassis serial number, all the modified information using the CMP shell is reverted back to factory default values. This means that the username/password database would be erased and the default username/password is in effect. The domain name used in service advertisement reverts to the chassis serial number plus slot ID.

Limitations

These are the limitation of CMP:

- CMP supports only SSH service.
- Only one SSH session has console, auxiliary or system admin port.
- CMP does not support software image upgrade, pie or VM downloads.
- IPv6 link local address is preferred by Avahi application rather than the IPv6 global address.
- There is no authentication performed on users logging into the CMP shell.
- Warm reload causes loss of CMP SSH session only in A9K-RSP880-TR/SE or A99-RP2-TR/SE.

Zero Configuration Networking

CMP configures the network devices using zero-configuration networking model and eliminates the need to have serial terminal servers. The zero-configuration networking enables:

- automatic IP address selection for network device—If a network device does not have an IP address assigned to it, then zero-configuration networking supports DHCP to obtain IPv6 Stateless Address Autoconfiguration (SLAAC), IPv4 and IPv6 addresses. The CMP port when connected to a IPv6 network obtains a link local address and also IPv6 global auto address based on IPv6 SLAAC.
- automatic domain name resolution and distribution of computer host names—The zero-configuration networking implements multicast DNS (mDNS). mDNS allows a network device to select a domain name in the local namespace and then broadcast that name using a special multicast IP address, allowing other devices on the network to connect to it by name instead of by numbered IP address. This eliminates the need to configure a DNS server.
- automatic location of network services through DNS service discovery—The zero-configuration networking enables a network device to use standard DNS queries to discover devices registered on the

network that are broadcasting the services that they provide. This eliminates the need to set up a directory server.

These are the zero-configuration networking applications that are supported:

- For Windows and MAC OS—Bonjour
- For Linux OS—Avahi

CPAK

CPAKs are the Cisco's innovation for 100G pluggable optics, which is built with the industry leading smallest form factor, in full compliant with IEEE802.3ae specification for 100GE-SR10, -LR4, and can interoperate with all IEEE 802.3ba compliant CFP-SR10 or CFP-LR4 100G optics.

The key new functionality is that CPAK variants are being constructed that represent 10 x 10GE ports. A single physical port on the linecard needs to instantiate multiple breakout Ethernet interfaces, very much similar to serial interface channelization.

Modes Supported on CPAKs

This table clearly lists the modes supported with the relevant PID:

CPAK (PID)	Modes Supported
CPAK-100G-SR10	100 GE, 10 GE, 40 GE
CPAK-100G-LR	100 GE
CPAK-10X10G-LR	10 GE.
CPAK-100G-ER4L	100 GE

The standard R/S/I/P format is 4-tuple. 5-tuple interfaces are represented as - R/S/I/P/SP. P is the CPAK port and SP indicates the breakout port. A CPAK which is configured as 5 tuple after executing the **breakout** command can be configured as 0x10G configuration. A CPAK, without the breakout mode can only be configured as 100G, represents a 4 tuple configuration. The default interface type is HundredGigE. If there is no configuration, then Hundred GigE interface would be created for the CPAK ports.

Configuring Breakout

This task enables the user to configure the breakout option.

SUMMARY STEPS

1. **configure**
2. **hw-module location preconfigure** *location* **port breakout** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**hw-module location preconfigure** *location* **port breakout** *interface* **Example:** `RP/0/RSP0/CPU0:router (config) #` **`hw-module location 0/0/CPU0 port 0 breakout 10x TenGigE`**	Configures the breakout option. **Note** The optional keyword, **preconfigure** enables the user to preconfigure breakout on an empty slot. SR10 CPAK can operate in the following modes - 1x100GE or 10x10GE. 1x100GE is the default option. 10x10 CPAK can also support 10x10GE.

Power saving mode

8x100GE Line card consists of 4 Slices (0,1,2,3). Each slice has two physical ports. Slice-1, 2 and 3 can be configured into power save mode. Power save option is not applicable to Slice-0. Use the **hw-module power saving** command to configure the required slice to power saving mode.

Once a slice is configured in the power saving mode, the interfaces will be deleted and hence all traffic passing through the interfaces will be dropped.

Table 16: Slice-Port mapping table

Slice 1	Ports 2,3
Slice 2	Ports 4,5
Slice 3	Ports 6,7

To configure the power save option

This task enables the user to configure the power save option.

SUMMARY STEPS

1. **admin**
2. **configure**
3. **hw-module power saving location** *location* **slice** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**configure**	

	Command or Action	Purpose
Step 3	**hw-module power saving location** *location* **slice** *number* **Example:** `RP/0/RSP0/CPU0:router (admin-config) #` **`hw-module power saving location 0/1/CPU0 slice 3`**	Configures the power saving option for the specified slice. The available options are Slice1, 2, 3. **Note** Power save option is not applicable for Slice 0.

What to do next

Use the **show plat slices** command to get the status of the slices.

Configuring Breakout on a 400GE Port

Table 17: Feature History Table

Feature Name	Release Information	Description
Breakout in a 400GE Port	Release 7.4.1	You can configure breakout on a 400GE port for use the QSFP-28 transceiver module in a 1x100G mode. This feature is supported on Cisco ASR 9900 5th Generation 4Tb line cards (A99-10X400GE-X-SE/TR).
Configuring 4x100GE Breakout for ASR 9000 2T Combo Line Card (A9K-20HG-FLEX) and 800G Combo Line Card (A9K-8HG-FLEX)	Release 7.4.1	This release introduces support for 4x100 breakout on the ASR 9000 2T Combo Line Card (A9K-20HG-FLEX) and 800G Combo Line Card (A9K-8HG-FLEX). You can configure 4x100G breakout on a QSFP-DD port and use it as four individual 100G interfaces.

1x100GE Breakout

You can configure breakout on a 400GE port for use the QSFP-28 transceiver module in a 1x100G mode. This feature is supported on Cisco ASR 9900 5th Generation 4Tb line cards (A99-10X400GE-X-SE/TR). As part of this enhancement, 400GE ports support these QSFP28 optics:

- QSFP-100G-SR4-S
- QSFP-100G-PSM4-S
- QSFP-100G-CWDM4-S
- QSFP-100G-FR-S
- QSFP-100G-LR4-S
- QSFP-100G-ER4L-S
- QSFP-40/100-SRBD
- QSFP-100G-SM-SR

• QSFP-100G-LR-S

This task enables you to configure the breakout option:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# hw-module location 0/2/CPU0 port 1 breakout 1xHundredGige
```

Here's the **show** command output indicating the breakout configuration:

```
RP/0/RSP0/CPU0:router# show ipv4 interface brief | include Hun
Tue May  4 07:59:56.996 UTC
FourHundredGigE0/1/0/2         unassigned       Down             Down     default
HundredGigE0/2/0/0             198.127.6.1       Up               Up       vpn6
HundredGigE0/2/0/1             198.127.12.1      Up               Up       vpn12
HundredGigE0/2/0/2             198.127.10.1      Up               Up       vpn10
HundredGigE0/2/0/3             198.127.11.1      Up               Up       vpn11
HundredGigE0/2/0/4             198.127.9.1       Up               Up       vpn9
HundredGigE0/2/0/5             198.127.5.1       Up               Up       vpn5
HundredGigE0/2/0/6             198.127.1.1       Up               Up       vpn1
HundredGigE0/2/0/7             198.127.7.1       Up               Up       vpn7
HundredGigE0/2/0/8             198.127.4.1       Up               Up       vpn4
```

Here's the **show** command output indicating the 100G optics plugged in the QSFP-DD port number 2:

```
RP/0/RSP0/CPU0:router#show inventory loc 0/2
Tue May  4 07:58:58.461 UTC
NAME: "0/2", DESCR: "ASR 9900 10-port 400GE SE linecard"
PID: A99-10X400GE-X-SE , VID: V00, SN: FOC2423NXUH

NAME: "HundredGigE0/2/0/0", DESCR: "100GE-SR4-S QSFP Module"
PID: QSFP-100G-SR4-S   , VID: V03, SN: INL23332207

NAME: "HundredGigE0/2/0/1", DESCR: "100GE-LR4-S QSFP Module"
PID: QSFP-100G-LR4-S   , VID: V01, SN: FNS213210US

NAME: "HundredGigE0/2/0/2", DESCR: "100GE-SM-SR QSFP Module"
PID: QSFP-100G-SM-SR   , VID: V02, SN: FNS22151794

NAME: "HundredGigE0/2/0/3", DESCR: "100GE-FR-S QSFP Module"
PID: QSFP-100G-FR-S    , VID: V01, SN: INL24080848

NAME: "HundredGigE0/2/0/4", DESCR: "100GE-CWDM4-S QSFP Module"
PID: QSFP-100G-CWDM4-S , VID: V01, SN: JFQ210530CN

NAME: "HundredGigE0/2/0/5", DESCR: "100GE-ER4L QSFP Module"
PID: QSFP-100G-ER4L-S  , VID: V01, SN: FLJ2150J00V

NAME: "HundredGigE0/2/0/6", DESCR: "100GE-40/100-SRBD QSFP Module"
PID: QSFP-40/100-SRBD  , VID: V01, SN: FOF2145N0LA

NAME: "HundredGigE0/2/0/7", DESCR: "100GE-PSM4 QSFP Module"
PID: QSFP-100G-PSM4-S  , VID: V01, SN: CVR220500MB

NAME: "HundredGigE0/2/0/8", DESCR: "100GE-LR-S QSFP Module"
PID: QSFP-100G-LR-S    , VID: V01, SN: FBN2415B004
```

4x100GE Breakout

The QSFP-DD modules support 400Gbps. You can configure 4x100G breakout on a QSFP-DD port and use it as four individual 100G interfaces.

This feature is supported on the following variant of Cisco ASR 9000 Series Line Cards:

- A99-10X400GE-X-SE
- A99-10X400GE-X-TR
- A9K-20HG-FLEX-SE
- A9K-20HG-FLEX-TR
- A9K-8HG-FLEX-SE
- A9K-8HG-FLEX-TR

This example shows how to enable 4x100GE breakout:

```
RP/0/RSP0/CPU0:router(config)# hw-module location 0/8/CPU0 port 0 breakout 4xHundredGigE
```

The following output shows the breakout configuration:

```
RP/0/RP0/CPU0:router# show ipv4 interfaces brief | i 0/8/0/0/

HundredGigE0/8/0/0/0          10.0.9.1       Up              Up       vpn9
HundredGigE0/8/0/0/1          10.0.8.1       Up              Up       vpn8
HundredGigE0/8/0/0/2          10.0.7.1       Up              Up       vpn7
HundredGigE0/8/0/0/3          10.0.2.1       Up              Up       vpn2

RP/0/RP0/CPU0:router# show interface hundredGigE 0/8/0/0/*

HundredGigE0/8/0/0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is HundredGigE, address is 7c21.0e33.35f0 (bia 7c21.0e33.35f0)
  Layer 1 Transport Mode is LAN
  Description: noshut-interface
  Internet address is 10.0.9.1/24
  MTU 9216 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 100000Mb/s, 400GBASE-DR4, link type is force-up
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
  loopback not set,
  Last link flapped 08:07:28
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     18400 packets input, 4669038 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 1 broadcast packets, 18397 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     19128 packets output, 4732388 bytes, 0 total output drops
     Output 1 broadcast packets, 19125 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

```
     1 carrier transitions

HundredGigE0/8/0/0/1 is up, line protocol is up
  Interface state transitions: 1
  Hardware is HundredGigE, address is 7c21.0e33.35f1 (bia 7c21.0e33.35f1)
  Layer 1 Transport Mode is LAN
  Description: noshut-interface
  Internet address is 10.0.8.1/24
  MTU 9216 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 100000Mb/s, 400GBASE-DR4, link type is force-up
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
  loopback not set,
  Last link flapped 08:07:28
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     18408 packets input, 4677424 bytes, 5 total input drops
     0 drops for unrecognized upper-level protocol
     Received 1 broadcast packets, 18405 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     19118 packets output, 4725316 bytes, 0 total output drops
     Output 1 broadcast packets, 19115 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions

HundredGigE0/8/0/0/2 is up, line protocol is up
  Interface state transitions: 1
  Hardware is HundredGigE, address is 7c21.0e33.35f2 (bia 7c21.0e33.35f2)
  Layer 1 Transport Mode is LAN
  Description: noshut-interface
  Internet address is 10.0.7.1/24
  MTU 9216 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 100000Mb/s, 400GBASE-DR4, link type is force-up
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
  loopback not set,
  Last link flapped 08:07:28
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     18446 packets input, 4676812 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 1 broadcast packets, 18443 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     19184 packets output, 4729960 bytes, 0 total output drops
     Output 1 broadcast packets, 19181 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions

HundredGigE0/8/0/0/3 is up, line protocol is up
  Interface state transitions: 1
```

```
Hardware is HundredGigE, address is 7c21.0e33.35f3 (bia 7c21.0e33.35f3)
Layer 1 Transport Mode is LAN
Description: noshut-interface
Internet address is 10.0.2.1/24
MTU 9216 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)
   reliability 255/255, txload 0/255, rxload 0/255
Encapsulation ARPA,
Full-duplex, 100000Mb/s, 400GBASE-DR4, link type is force-up
output flow control is off, input flow control is off
Carrier delay (up) is 10 msec
loopback not set,
Last link flapped 08:07:28
ARP type ARPA, ARP timeout 04:00:00
Last input 00:00:00, output 00:00:00
Last clearing of "show interface" counters never
5 minute input rate 1000 bits/sec, 1 packets/sec
5 minute output rate 1000 bits/sec, 1 packets/sec
   18444 packets input, 4677024 bytes, 0 total input drops
   0 drops for unrecognized upper-level protocol
   Received 1 broadcast packets, 18441 multicast packets
            0 runts, 0 giants, 0 throttles, 0 parity
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   19170 packets output, 4734446 bytes, 0 total output drops
   Output 1 broadcast packets, 19167 multicast packets
   0 output errors, 0 underruns, 0 applique, 0 resets
   0 output buffer failures, 0 output buffers swapped out
   1 carrier transitions
```

Reloading, Shutting Down, or Power Cycling a Node

Use the commands described in this section to reload the Cisco IOS XR software on the active RSP or on any specified node in the system. This section also describes the commands used to administratively shut down a node and power a node on or off.

Table 18: Commands to Reload, Shut Down, or Power Cycle a Node

Command	Description
hw-module location *node-id* **power disable**	This command administratively turns the power off for a node. It is entered in administration configuration mode. The changes do not take effect until you enter the **commit** command. To power on a node, use the **no** form of this command. **Note** This command cannot be used to disable power on the RSP from which the command is entered.
hw-module location *node-id* **reload**	This command works in EXEC mode and reloads the Cisco IOS XR software on a specific node or all nodes. To specify all nodes, enter the **all** keyword in place of the *node-id* argument. The node reloads with the current running configuration and active software set for that node.

Command	Description
hw-module shutdown location *node-id*	This command must be entered in the configuration mode and administratively shuts down the specified node. Nodes that are shut down still have power but cannot load or operate Cisco IOS XR software. To return a node to the up state, use the **no** form of this command. **Note** This command cannot be used to shut down the RSP from which the command is entered.
hw-module unshut location *node-id*	This command must be entered in the configuration mode. This command is used to administratively bring up the specified node.

Note When you use the **hw-module shutdown location** *node-id* command to a line card, you must wait until the configuration is applied before removing the line card. Removal of the line card before the shutdown may result in a hardware issue.

Reloading the Active RSP

The **reload** command causes the active RSP to reload the Cisco IOS XR software according to the configuration register setting. This setting determines how the active RSP acts when reloaded.

This section contains instructions to reload the Cisco IOS XR software and return to EXEC mode. For instructions to use the **reload** command for entering ROM Monitor bootstrap mode, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.

Caution Because the **reload** command causes the active RSP to go off line and either reload the Cisco IOS XR software or enter ROM Monitor mode, the router experiences a loss of service unless a redundant standby RSP is installed and in "ready" state. To display the status of the standby RSP, use the **show redundancy** command in EXEC mode.

SUMMARY STEPS

1. **show redundancy**
2. **admin**
3. **show variables boot**
4. (Optional) **config-register** *register-value*
5. **admin**
6. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show redundancy**	Displays the RSP redundancy status.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router# show redundancy`	• If a standby RSP is in "ready" redundancy state, the **reload** command also causes the router to gracefully fail over to the standby RSP.
Step 2	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 3	**show variables boot** **Example:** `RP/0/RSP0/CPU0:router(admin)# show variables boot`	Displays the configuration register setting. • Enter this command in administration EXEC mode. • For normal operations, the configuration register setting is 0x102 or 0x2102, which causes the active RSP to reload the Cisco IOS XR software. • Verify that the configuration register setting is 0x102 or 0x2102. If it is not, complete Step 4, on page 140 to reset the configuration register to 0x102 or 0x2102. **Note** For instructions on how to enter ROM Monitor bootstrap mode, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.
Step 4	(Optional) **config-register** *register-value* **Example:** `RP/0/RSP0/CPU0:router(admin)# config-register 0x102`	Sets the configuration register to the respective value. This step is necessary only if the register is not set to the respective value (0x102 or 0x2102) in the running configuration. You can use either 0x102 or 0x2102. Both these values specify the same functionality, as bit 13 in 0x2102 is not significant for Cisco IOS XR software.
Step 5	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 6	**reload** **Example:** `RP/0/RSP0/CPU0:router# reload`	Reloads the active RSP according to the configuration register setting. • If the setting is 0x102 or 0x2102 , then the RSP reloads the Cisco IOS XR software. • If the standby RSP is in "ready" redundancy state, the router switches over to the standby RSP. • If a standby RSP is not installed or not in a "ready" state, the router experiences a loss of service while the active RSP is reloading the Cisco IOS XR software.

Flash Disk Recovery

When an RSP is power cycled or experiences an ungraceful reset, the boot disk (PCMCIA flash disk used to boot the card) may experience a file-system corruption. If this occurs, an error message is displayed and the RSP fails to boot. The corrupted flash disk is automatically reformatted and the Cisco IOS XR software is restored from the designated system controller (DSC) for the system.

For example, if a flash disk for an RSP is corrupted, the RP fails to boot and the following error message is displayed:

```
#########################################################
              Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706

Cisco IOS XR Software for the Cisco XR Cisco ASR 9000 Series Router-mbirp,
Copyright (c) 2009 by Cisco Systems, Inc.
Unable to mount /disk0:, filesystem is corrupted.
Check fsck log at /tmp/chkfs_fd0.log
init: special_commands:wait for disk0: failed
```

If this occurs, then the flash disk is automatically reformatted and the Cisco IOS XR software is restored to the flash disk.

Note If the flash disk is badly damaged and cannot be reformatted, the disk must be replaced.

If the corrupted flash disk is the DSC, then the router fails over to the standby DSC. If no standby DSC is installed, then the system fails to boot.

Using Controller Commands to Manage Hardware Components

The **controller** , **controllers** , and **show controllers** commands are used to manage and display settings for various hardware components, including the switch fabric management, Ethernet control plane, and interface manager. These commands are primarily diagnostic and related to driver-level details. The information available with these commands varies widely and is hardware specific.

For information on the use of these commands, see *Interface and Hardware Component Command Reference for Cisco ASR 9000 Series Routers*.

Formatting Hard Drives, Flash Drives, and Other Storage Devices

To format a storage device on the router, use the **format** command in EXEC mode.

> **Caution** Formatting a storage device deletes all data on that device.

The following command syntax is used:

format *filesystem:* [*options*]

Table 19: format command Syntax Description, on page 142 describes the **format** command syntax.

Table 19: format command Syntax Description

Variable	Description
filesystem	Specifies the memory device to format. The supported file systems are: • **bootflash:** • **compactflash:** • **configflash:** • **harddisk:** • **harddiska:** • **disk0:** • **disk1:** Enter **format ?** to see the devices supported on your router.
options	Enter **format** *filesystem:* **?** to see the available options. For more information, see *System Management Command Reference for Cisco ASR 9000 Series Routers*.

In the following example, the **format** command is used to format the hard disk:

```
RP/0/RSP0/CPU0:router# format harddisk:
```

Removing and Replacing Cards

This section describes card replacement issues and procedures.

Removing Line Cards

Line cards are designed for online insertion and removal (OIR). A line card is a single card that contains all service processing functions and physical line interfaces.

The OIR feature allows you to remove and replace cards without removing power to the card or chassis. Removing a card interrupts all traffic passing through the card, but it does not remove the card configuration.

When you remove a card, the configuration remains for all interfaces, but the interfaces do not appear in the output of the **show interfaces** command. You can view interface configurations by entering the **show running-config** command. The following example shows how the configuration appears when a card is removed:

```
RP/0/RSP0/CPU0:router# show running-config

  Building configuration...
  hostname router
  router ospf 3269
   area 0
    interface POS0/3/0/0
     cost 20
  !
  interface preconfigure POS0/3/0/0
   ipv4 address 10.10.50.1 255.255.255.0
  !
  interface preconfigure POS0/3/0/1
   description POS0/3/0/1
   shutdown
  !
  interface preconfigure POS0/3/0/2
   description POS0/3/0/2
   shutdown
  !
  interface preconfigure POS0/3/0/3
   description POS0/3/0/3
   shutdown
  !
```

In this example, the line card in slot 3 is removed, and the interface configuration for all four interfaces changes to "interface preconfigure." However, the "router ospf" reference to a slot 3 interface does not change. If you replace a line card with another line card that uses the same media type and port count, the configuration becomes active on the replacement card.

To remove the configuration for a slot after a card is removed, use the **no interface preconfigure** command to remove all interface configuration statements for that card in the running configuration. In addition, search the configuration for any references to the removed interfaces, such as the "router ospf" reference to slot 3 in the preceding example.

To remove the configuration for a slot when a card is installed, use the **no interface** command to remove all interface configuration statements for that card in the running configuration. In addition, search the configuration for any references to the removed interfaces.

Each line card supports a specific media type (Packet over SONET/SDH [POS] or Ethernet, for example) and port count. If you replace a line card with one that supports a different media type or port count, you should review the configuration and revise it to support the replacement line card.

Replacing a Line Card with the Same Media Type and Port Count

When you replace a line card or PLIM with a card that is of the same media type and has the same port count as the replaced card, the guidelines in the Removing Line Cards , on page 142 apply. Because the replacement card is of the same media type and port count, no special procedures are required for card removal and replacement.

Replacing a Line Card with the Same Media Type and a Different Port Count

When you replace a line card with a card that is of the same media type with a different port count, the guidelines in Removing Line Cards , on page 142 apply.

If the new card has a greater port count than the replaced card, the configuration applies to the corresponding lower port numbers, and the ports that did not exist on the replaced card have no configuration and come up in the shutdown state.

If the new card supports fewer ports, the existing configuration for the corresponding number of ports on the new card set is applied. The previous configuration for the removed ports remains in interface preconfigure state, as shown in the following example:

```
RP/0/RSP0/CPU0:router# show running-config

  Building configuration...
  hostname rtp-gsr1
  interface POS0/3/0/0
   ipv4 address 10.10.50.1 255.255.255.0
  !
  interface preconfigure POS0/3/0/1
   description POS0/3/0/1
   shutdown
  !
  interface preconfigure POS0/3/0/2
   description POS0/3/0/2
   shutdown
  !
  interface preconfigure POS0/3/0/3
   description POS0/3/0/3
   shutdown
  !
```

In the preceding example, a four-port card has been replaced with a single-port card. The configuration from port 1 on the four-port card is applied to the single port on the replacement card, and the remaining port configurations change to “interface preconfigure.” To remove the configuration for the missing interfaces, use the **no interface preconfigure** command. In addition, search for and remove any configuration references to the removed interfaces.

Whenever you replace a line card with the same media type and a different port count, review the running configuration in the router and revise the configuration as necessary.

Replacing a Line Card or PLIM with a Different Media Type

When you replace a line card or PLIM with a card that is of a different media type (for example, if you replace a POS PLIM with an Ethernet PLIM), the guidelines in Removing Line Cards , on page 142 apply. Review the running configuration in the router and revise the configuration as necessary for the new media type.

Examples: Breakout and Power saving options

The following are the examples for the **power save** and **breakout** options:

Power saving mode

Configuring the power saving option:

```
admin
```

```
 config
  hw-module power saving location 0/0/CPU0 slice 3
 !
```

```
show platform slices
Line Card       Slice   Config          Status
0/0/CPU0        0       Power on        Completed
                1       Power on        Completed
                2       Power on        Completed
                3       Power saving    Completed
```

Breakout option

Configuring the breakout option:

```
config
 hw-module location 0/0/CPU0 port 0 breakout 10xTenGigE
!
```

show command output indicating the breakout ports:

```
RP/0/RSP0/CPU0:TD02#show ipv4 interface brief | include Hun
Sun Sep  7 15:59:33.446 PST
HundredGigE0/0/0/0             34.34.34.2      Down                   Down
HundredGigE0/0/0/1             100.0.1.1       Up                     Up
HundredGigE0/0/0/2             unassigned      Up                     Up
HundredGigE0/0/0/3             unassigned      Up                     Up
HundredGigE0/0/0/4             unassigned      Shutdown               Down
HundredGigE0/0/0/5             unassigned      Shutdown               Down
HundredGigE0/0/0/6             unassigned      Shutdown               Down
HundredGigE0/0/0/7             unassigned      Shutdown               Down


RP/0/RSP0/CPU0:router(config)#hw-module location 0/0/CPU0 port 2 breakout 10xTenGigE
RP/0/RSP0/CPU0:router(config)#commit

RP/0/RSP0/CPU0:router#show ipv4 interface brief | include Ten
TenGigE0/0/0/2/0               unassigned      Shutdown               Down
TenGigE0/0/0/2/1               unassigned      Shutdown               Down
TenGigE0/0/0/2/2               unassigned      Shutdown               Down
TenGigE0/0/0/2/3               unassigned      Shutdown               Down
TenGigE0/0/0/2/4               unassigned      Shutdown               Down
TenGigE0/0/0/2/5               unassigned      Shutdown               Down
TenGigE0/0/0/2/6               unassigned      Shutdown               Down
TenGigE0/0/0/2/7               unassigned      Shutdown               Down
TenGigE0/0/0/2/8               unassigned      Shutdown               Down
TenGigE0/0/0/2/9               unassigned      Shutdown               Down
```

Proactive Line Card Shut Down

The proactive line card shutdown feature enables powering down line cards automatically when the power that is drawn by the router exceeds configured threshold. The sequence of powering down LCs is based on the shutdown priorities that are assigned to them. The LC, however, does not boot automatically even after the router power draw is back to normal below the shutdown threshold. Therefore, you must manually bring up the LC by either reloading or OIR method.

Instead of provisioning more power to the router in worst case power draw scenarios, this feature helps in saving power by powering down the LC.

As part of this feature, you must configure two threshold values:

- Syslog Threshold—This value serves as a syslog warning threshold. If the router power draw exceeds the syslog threshold, then a warning error message is captured at the console. This warning message facilitates the user to take any preventive action.
- Shutdown Threshold—This value triggers the shutdown of line cards. If the router power draw exceeds the shutdown threshold, then the line cards are shutdown based on the priorities that are assigned to them. The system monitors the power draw for every 10 seconds.

Shut Down Priorities for Powering Down the LC

You can configure a shutdown priority value of 0 to 19. A line card with lower value has the highest priority. By default, a priority of 20 is assigned to all the LC in the router.

When two or more LCs have equal priorities that are assigned, then the slot number takes precedence in the priority calculation. A lower slot number has the highest priority. For instance, when two LCs at slot 0 and slot 19 have the priority set as 10, then LC in slot 0 has higher priority than the LC in slot 19.

Proactive Line Card Shut Down Implementation Consideration

Consider the following points while configuring proactive line card shut down feature:

- Shutdown threshold must be greater than the current system power draw.
- Shutdown threshold must be greater than the syslog threshold.
- Shutdown threshold must not be less than 3500 Watts.
- In Cisco IOS XR 32-bit OS, the priority of a LC is checked only when the LC is in **IOS XR RUN** state.
- In Cisco IOS XR 64-bit OS, the priority of a LC is checked only when the LC is in **Operational** state.

Configure Proactive Line Card Shut Down

Cisco IOS XR 32-bit

Configuring proactive line card shutdown includes:

- Assigning priorities to the line cards
- Configuring syslog threshold
- Configuring shutdown threshold

In this example, syslog threshold of 5000 W and shutdown threshold of 6000 W is configured along with the LC priorities:

```
config
power budget enforcement progressive
priority 1
  location 0/1/CPU0
  location 0/5/CPU0
!
priority 5
```

```
  location 0/4/CPU0
!
priority 11
  location 0/2/CPU0
!
priority 20
  location 0/3/CPU0
!
syslog-threshhold 5000 W
shutdown-threshhold 6000 W
```

The following error message is seen when power draw exceeds the shutdown threshold:

```
RP/0/RP0/CPU0:Mar  8 11:42:00.146 : shelfmgr[406]: %PLATFORM-SHELFMGR-1-INRESET_ALARM :
Power off node 0/10/CPU0 due to multiple critical alarms, putting into IN_RESET state
RP/0/RP0/CPU0:Mar  8 11:42:10.948 : envmon[209]: %PLATFORM-ENVMON-2-PWR_EXCEEDED_SHUTDOWN
: Slot 0/3/CPU0 priority 20 is being shutdown,current power usage 6746 W exceeds the
configured threshold of 6000 W
```

Cisco IOS XR 64-bit

Configuring proactive line card shutdown includes:

- Configuring syslog threshold
- Configuring shutdown threshold
- Assigning priorities to the line cards

In this example, syslog threshold of 3300 W and shutdown threshold of 3500 W is configured along with LC priorities:

```
sysadmin config
 power-mgmt progressive enable
  syslog-threshold 3300        /* syslog generated when power crosses this value */
  shutdown-threshold 3500      /* LCs shut down based on priority once power draw crosses
this limit */
  priority location 0/0 10    /* Priority assigned for each LC */
  !
  priority location 0/1 5
  !
  priority location 0/4 4
  !
  priority location 0/6 2
  !
  priority location 0/7 1
  !
!
```

The following error message is seen when power draw exceed the syslog threshold:

```
0/RSP0/ADMIN0:Feb 22 11:44:38.566 UTC: envmon[4202]: %PWR_MGMT-ENVMON-3-PWR_EXCEEDED_WARN
:
 Chassis power usage 3448 W has exceeded the configured warning threshold of 3300 W
```

The following error message is seen when power draw exceeds the shutdown threshold:

```
0/RSP0/ADMIN0:Feb 22 11:44:38.567 UTC: envmon[4202]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM_MAJOR :Node shutdown by Progressive power-mgmt mode :DECLARE :0/0:
Node priority 10, Chassis power draw 3570 W exceeded shutdown threshold 3500 W
```

Advanced Power Management

Advanced power management feature enables powering down the unused line card slices.

This feature helps to manage power consumption, as the slices that do not have any services enabled are power down. Later these slices can be powered when a new service is enabled on them.

This feature is supported on the Cisco ASR 9000 4th Generation Ethernet line cards.

Configuring Advance Power Management

This procedure shows how to configure advance power management.

In this example these slices are powered down:

- slice 0, and 7 of the line card in the node 0 location
- slice 3, and 6 of the line card in the node 1 location

You should reload the line card for the configuration changes to take effect.

```
Router# configure
Router (config)# hw-module location 0/0/CPU0 slice 0 power-down
Router (config)# hw-module location 0/0/CPU0 slice 7 power-down
Router (config)# hw-module location 0/1/CPU0 slice 3 power-down
Router (config)# hw-module location 0/1/CPU0 slice 6 power-down
Router (config)# commit
Router (config)# end
Router # admin
Router (sysadmin-vm)# hw-module location 0/0 reload
Router (sysadmin-vm)# hw-module location 0/1 reload
```

Running Configuration

```
config
hw-module location 0/0/CPU0 slice 0 power-down
hw-module location 0/0/CPU0 slice 7 power-down
hw-module location 0/1/CPU0 slice 3 power-down
hw-module location 0/1/CPU0 slice 6 power-down
```

Verification

```
Router# show apm psm status

PSM Status
----------
PSM Client Status
        DIAG0:          Not registered
        DIAG1:          Registered
        0/1 PSA:        Registered

LC Status
---------
Line Card       Slice   Config  Status          DIAG0           DIAG1           PSA
```

```
0/0/CPU0       0      On      Completed      Not registered  Completed      Not present
               1      On      Completed      Not registered  Completed      Not present
               2      On      Completed      Not registered  Completed      Not present
               3      On      Completed      Not registered  Completed      Not present
               4      On      Completed      Not registered  Completed      Not present
               5      On      Completed      Not registered  Completed      Not present
               6      On      Completed      Not registered  Completed      Not present
               7      On      Completed      Not registered  Completed      Not present

0/1/CPU0       0      On      Completed      Not registered  Completed      Completed
               1      On      Completed      Not registered  Completed      Completed
               2      On      Completed      Not registered  Completed      Completed
               3      On      Completed      Not registered  Completed      Completed
               4      On      Completed      Not registered  Completed      Completed
               5      On      Completed      Not registered  Completed      Completed
               6      On      Completed      Not registered  Completed      Completed
               7      On      Completed      Not registered  Completed      Completed

New configuration after line card reboots
-----------------------------------------
Line Card        Slice   New Config
0/0/CPU0         0       Down
                 7       Down
0/1/CPU0         3       Down
                 6       Down
```

Overview of Erase and Wipeout Disk Memory

Below two methods are used to delete the data from a RSP and line card. These methods are used based on your requirements:

- Erase Disk Memory
- Wipe Out Disk Memory

Erase Disk Memory

The Erase Disk Memory operation clears the disk memory of RSPs and line cards. However, the deleted data is recoverable using recovery tools. The erase disk memory operation can be performed for quick sanitization of the card before reusing it in another device within the control space of your network or organization.

How to Erase Disk Memory

Erasing disk memory operation uses zapdisk feature to erase the disk memory from the RSP and line card.

Erasing disk memory is done in three steps. First, you enable the zapdisk feature, later identify the card where zapdisk is supported. Next, activate the zapdisk operation on the card:

1. Enable zapdisk feature on the router.

   Example:

   ```
   sysadmin-vm# zapdisk set
   ```

2. Find out the card location where the zapdisk feature is supported using the **show zapdisk locations** command.

   Example:

   ```
   Router# show zapdisk locations

   0/RSP1      Fully qualified location specification
   0/7         Fully qualified location specification
   0/4         Fully qualified location specification
   all         all locations
   ```

3. Start the zapdisk operation on a specific node location or all node locations to erase disk memory.

   Note You can run the zapdisk operation on all RSPs and line cards except the active RSP where zapdisk service is running in an active role.

   After the zapdisk process is completed, the system clears all data and shuts down the card.

   This example runs the zapdisk operation on the node location 0/4:

   ```
   Router# zapdisk start location 0/4
   Action on designated location is in progress, more detail logs will be located in sysadmin
    at
   /misc/disk1/tftpboot/zapdisk.log once action is completed
   ```

Note After deleting the data, remove the card from the slot, and do not reload the card or the router. If you reload the card or the router without removing the card, the data is reloaded into the card.

In the event when you must return or trash a card, the data in the disk memory should be permanently deleted. Therefore, the erase disk memory feature is not advice. You should use the enhanced version of the erase disk memory feature called Wipe Out Disk Memory.

Wipe Out Disk Memory

The Wipe Out Disk Memory feature deletes data permanently from the disk memory of RSPs and line cards. The erased data is non-recoverable. We recommmend this action when you perform a return material authorization (RMA) of a card to prevent pilferage of sensitive data.

How to Wipe out Disk Memory

Wiping out disk memory actions are performed in the ROMMON mode. Generally to boot into ROMMON mode, the **config-register boot-mode rom-monitor** command is executed from the admin mode. However, the command is not available in Cisco IOS XR 64 bit OS. Therefore you must follow the below sequence to boot into ROMMON mode:

1. Reload the router
2. Break into the BIOS menu and select ROMMON
3. Wipe out disk memory in ROMMON

Reload the router

Before reloading the router, ensure that the redundant RP is disabled in dual-RP routers and console is connected:

```
sysadmin-vm:0_RSP0# hw-module location all reload
```

Break into the BIOS menu and select ROMMON

1. While the router boots, press CTRL+C to break into BIOS menu.
2. To enter into ROMMON mode, select the Boot to ROMMON option from the available boot options:

```
Please select the operating system and the boot device:
     1) Boot to ROMMON
     2) IOS-XR 64 bit Boot previously installed image
     3) IOS-XR 64 bit Mgmt Network boot using DHCP server
     4) IOS-XR 64 bit Mgmt Network boot using local settings (iPXE)
     (Press 'p' for more option)
Selection [1/2/3/4]: 1
Selected Boot to ROMMON , Continue ? Y/N: y

rommon 1 >
```

Wipe out disk memory in ROMMON

1. Go to Privilege Mode.

```
rommon > priv
```

2. Select the **hderase** option.

```
rommon > hderase
            SATA HD(0x4,0x0,0x0):
         Model    : <Model number>
    Serial No : <serial number>

     Sanitize Crypto Scramble Erase Supported
     Sanitize State : Idle

All the contents on this Drive will be Erased
Do you wish to continue?(Y/N)
Y
```

The data is permanently erased.

Upgrading the CPU Controller Bits

Use this procedure to upgrade the CPU controller bits on all nodes that are installed in the router or on a specific node.

SUMMARY STEPS

1. **admin**
2. **upgrade cpuctrlbits** {**all** | **location** *node-id*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**upgrade cpuctrlbits** {**all** | **location** *node-id*} **Example:** `RP/0/RSP0/CPU0:router(admin)# upgrade cpuctrlbits all`	Upgrades the CPU controller bits on all nodes in the router. Use the **location** *node-id* keyword and argument to upgrade the CPU controller bits on a specific node.

Examples

The following example shows how to upgrade the CPU controller bits on all nodes in a router:

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# upgrade cpucrtlbits all

Please do not power cycle, reload the router or reset any nodes until all upgrades are
completed.
Please check the syslog to make sure that all nodes are upgraded successfully.
If you need to perform multiple upgrades, please wait for current upgrade to be completed
before proceeding to another upgrade. Failure to do so may render the cards under upgrade
to be unusable.
```

Configuring Port Modes

This section describes how to configure the various port modes on a router, port expansion card, or a line card.

Table 20: Feature History Table

Feature Name	Release Information	Description
1GbE Port Mode on SFP+ and SFP28 Ports	Release 7.9.1	It is now possible to use the 10GbE SFP+ and SFP28 ports to transmit 1GbE traffic on the routers or line cards. The number of ports on which you can enable 1GbE port mode has been increased from the previous releases. The details of the number of ports on which you can configure 1GbE port mode on the Cisco ASR 9000 Series routers, port expansion card, and line cards is as follows: • Cisco ASR 9903 - 20 SFP+ ports • Cisco ASR 9902 - 24 SFP+ and 16 SFP28 ports • A9903-8HG-PEC - 40 SFP+ ports • A9K-4HG-FLEX-SE/TR - 24 SFP+ and 16 SFP28 ports • A99-4HG-FLEX-SE/TR - 24 SFP+ and 16 SFP28 ports • A9K-4HG-FLEX-FC - 24 SFP+ and 16 SFP28 ports • A99-4HG-FLEX-FC - 24 SFP+ and 16 SFP28 ports Use the **hw-module location breakout** and **hw-module location slice config mode** commands to enable 1GbE port mode configuration.

Configuring Port Mode in Cisco A9903-8HG-PEC

You can configure the SFP+ ports on the Cisco A9903-8HG-PEC (0.8T PEC card) in 10GbE, 25GbE, or 1GbE mode. By default, all ports are in 10GbE mode.

For more information on the ports, see the Installing Modules and Cables in the Chassis chapter in the *Cisco ASR 9000 Series Fixed-Port Routers Hardware Installation Guide*.

To configure a port, use the **hw-module location slice config mode** command.

The port mode configuration is per slice and not per port. Also, the 48 ports on Cisco A9903-8HG-PEC are available in slices 4 and 5. Each slice can be configured for 10GbE, 25GbE, or 1GbE port mode.

The number of ports available on the card depends on the port mode configured on each slice. This table shows the port mode, and available ports on Cisco A9903-8HG-PEC:

Table 21: Port Modes, and Available Ports on Cisco A9903-8HG-PEC

Port Mode	Number of Ports	Available Port Numbers
1GbE	48 24 on each slice	On slice 4: port numbers 0–23 On slice 5: port numbers 24–47
10GbE	48 24 on each slice	On slice 4: port numbers 0–23 On slice 5: port numbers 24–47
25GbE	32 16 on each slice	On slice 4: port numbers 0–15 On slice 5: port numbers 32–47

Configuration

This sample configuration shows how to configure all ports on Cisco A9903-8HG-PEC in 25GbE mode:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 slice 4 config-mode
4x25GE,4x25GE,4x25GE,4x25GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 slice 5 config-mode
4x25GE,4x25GE,4x25GE,4x25GE
RP/0/RP0/CPU0:ios(config)#commit
```

This sample configuration shows how to configure all ports on Cisco A9903-8HG-PEC in 1GbE mode:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 slice 4 config-mode
10x1GE,10x1GE,4x10GE,UNUSED
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 slice 5 config-mode
10x1GE,10x1GE,4x10GE,UNUSED
RP/0/RP0/CPU0:ios(config)#commit
```

By default, all ports are in 10GbE mode. To revert to the default mode, use **no** form of the `hw-module location <node> slice <number> config-mode` command. This sample shows how to revert slice 4 ports to the default 10GbE mode:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#no hw-module location 0/0/CPU0 slice 4 config-mode
4x25GE,4x25GE,4x25GE,4x25GE
RP/0/RP0/CPU0:ios(config)#commit
```

Verification

Use the **show interface brief** command to verify the ports in slice 4 and 5 configured in 25GbE mode:

```
RP/0/RP0/CPU0:ios#show interfaces brief
Intf     Intf        LineP      Encap MTU     BW
Name     State       State      Type (byte)  (Kbps)
--------------------------------------------------------------------------------
[SNIP]
TF0/0/1/0 admin-down admin-down ARPA 1514 25000000 /* slice 4 port numbers starts here */
TF0/0/1/1 admin-down admin-down ARPA 1514 25000000
TF0/0/1/2 admin-down admin-down ARPA 1514 25000000
TF0/0/1/3 admin-down admin-down ARPA 1514 25000000
```

```
TF0/0/1/4 admin-down admin-down ARPA 1514 25000000
TF0/0/1/5 admin-down admin-down ARPA 1514 25000000
TF0/0/1/6 admin-down admin-down ARPA 1514 25000000
TF0/0/1/7 admin-down admin-down ARPA 1514 25000000
TF0/0/1/8 admin-down admin-down ARPA 1514 25000000
TF0/0/1/9 admin-down admin-down ARPA 1514 25000000
TF0/0/1/10 admin-down admin-down ARPA 1514 25000000
TF0/0/1/11 admin-down admin-down ARPA 1514 25000000
TF0/0/1/12 admin-down admin-down ARPA 1514 25000000
TF0/0/1/13 admin-down admin-down ARPA 1514 25000000
TF0/0/1/14 admin-down admin-down ARPA 1514 25000000
TF0/0/1/15 admin-down admin-down ARPA 1514 25000000
TF0/0/1/32 admin-down admin-down ARPA 1514 25000000 /* slice 5 port numbers starts here */

TF0/0/1/33 admin-down admin-down ARPA 1514 25000000
TF0/0/1/34 admin-down admin-down ARPA 1514 25000000
TF0/0/1/35 admin-down admin-down ARPA 1514 25000000
TF0/0/1/36 admin-down admin-down ARPA 1514 25000000
TF0/0/1/37 admin-down admin-down ARPA 1514 25000000
TF0/0/1/38 admin-down admin-down ARPA 1514 25000000
TF0/0/1/39 admin-down admin-down ARPA 1514 25000000
TF0/0/1/40 admin-down admin-down ARPA 1514 25000000
TF0/0/1/41 admin-down admin-down ARPA 1514 25000000
TF0/0/1/42 admin-down admin-down ARPA 1514 25000000
TF0/0/1/43 admin-down admin-down ARPA 1514 25000000
TF0/0/1/44 admin-down admin-down ARPA 1514 25000000
TF0/0/1/45 admin-down admin-down ARPA 1514 25000000
TF0/0/1/46 admin-down admin-down ARPA 1514 25000000
TF0/0/1/47 admin-down admin-down ARPA 1514 25000000
```

This show command displays all 48 ports on Cisco A9903-8HG-PEC in 10GbE mode:

```
RP/0/RP0/CPU0:ios#show interfaces brief
Intf      Intf        LineP      Encap MTU     BW
Name      State       State      Type (byte)  (Kbps)
--------------------------------------------------------------------------------
Te0/0/1/0 admin-down admin-down ARPA 1514 10000000 /* slice 4 port numbers starts here  */
Te0/0/1/1 admin-down admin-down ARPA 1514 10000000
Te0/0/1/2 admin-down admin-down ARPA 1514 10000000
Te0/0/1/3 admin-down admin-down ARPA 1514 10000000
Te0/0/1/4 admin-down admin-down ARPA 1514 10000000
Te0/0/1/5 admin-down admin-down ARPA 1514 10000000
Te0/0/1/6 admin-down admin-down ARPA 1514 10000000
Te0/0/1/7 admin-down admin-down ARPA 1514 10000000
Te0/0/1/8 admin-down admin-down ARPA 1514 10000000
Te0/0/1/9 admin-down admin-down ARPA 1514 10000000
Te0/0/1/10 admin-down admin-down ARPA 1514 10000000
Te0/0/1/11 admin-down admin-down ARPA 1514 10000000
Te0/0/1/12 admin-down admin-down ARPA 1514 10000000
Te0/0/1/13 admin-down admin-down ARPA 1514 10000000
Te0/0/1/14 admin-down admin-down ARPA 1514 10000000
Te0/0/1/15 admin-down admin-down ARPA 1514 10000000
Te0/0/1/16 admin-down admin-down ARPA 1514 10000000
Te0/0/1/17 admin-down admin-down ARPA 1514 10000000
Te0/0/1/18 admin-down admin-down ARPA 1514 10000000
Te0/0/1/19 admin-down admin-down ARPA 1514 10000000
Te0/0/1/20 admin-down admin-down ARPA 1514 10000000
Te0/0/1/21 admin-down admin-down ARPA 1514 10000000
Te0/0/1/22 admin-down admin-down ARPA 1514 10000000
Te0/0/1/23 admin-down admin-down ARPA 1514 10000000
Te0/0/1/24 admin-down admin-down ARPA 1514 10000000 /* slice 5 port numbers starts here */
Te0/0/1/25 admin-down admin-down ARPA 1514 10000000
Te0/0/1/26 admin-down admin-down ARPA 1514 10000000
Te0/0/1/27 admin-down admin-down ARPA 1514 10000000
```

```
Te0/0/1/28 admin-down admin-down ARPA 1514 10000000
Te0/0/1/29 admin-down admin-down ARPA 1514 10000000
Te0/0/1/30 admin-down admin-down ARPA 1514 10000000
Te0/0/1/31 admin-down admin-down ARPA 1514 10000000
Te0/0/1/32 admin-down admin-down ARPA 1514 10000000
Te0/0/1/33 admin-down admin-down ARPA 1514 10000000
Te0/0/1/34 admin-down admin-down ARPA 1514 10000000
Te0/0/1/35 admin-down admin-down ARPA 1514 10000000
Te0/0/1/36 admin-down admin-down ARPA 1514 10000000
Te0/0/1/37 admin-down admin-down ARPA 1514 10000000
Te0/0/1/38 admin-down admin-down ARPA 1514 10000000
Te0/0/1/39 admin-down admin-down ARPA 1514 10000000
Te0/0/1/40 admin-down admin-down ARPA 1514 10000000
Te0/0/1/41 admin-down admin-down ARPA 1514 10000000
Te0/0/1/42 admin-down admin-down ARPA 1514 10000000
Te0/0/1/43 admin-down admin-down ARPA 1514 10000000
Te0/0/1/44 admin-down admin-down ARPA 1514 10000000
Te0/0/1/45 admin-down admin-down ARPA 1514 10000000
Te0/0/1/46 admin-down admin-down ARPA 1514 10000000
Te0/0/1/47 admin-down admin-down ARPA 1514 10000000
```

This show command displays ports in slice 4 configured in 25GbE mode and slice 5 in 10GbE mode:

```
RP/0/RP0/CPU0:ios#show interfaces brief
Intf      Intf        LineP      Encap MTU     BW
Name      State       State      Type (byte) (Kbps)
--------------------------------------------------------------------------------
TF0/0/1/0 admin-down admin-down ARPA 1514 25000000  /* slice 4 port numbers starts here */
TF0/0/1/1 admin-down admin-down ARPA 1514 25000000
TF0/0/1/2 admin-down admin-down ARPA 1514 25000000
TF0/0/1/3 admin-down admin-down ARPA 1514 25000000
TF0/0/1/4 admin-down admin-down ARPA 1514 25000000
TF0/0/1/5 admin-down admin-down ARPA 1514 25000000
TF0/0/1/6 admin-down admin-down ARPA 1514 25000000
TF0/0/1/7 admin-down admin-down ARPA 1514 25000000
TF0/0/1/8 admin-down admin-down ARPA 1514 25000000
TF0/0/1/9 admin-down admin-down ARPA 1514 25000000
TF0/0/1/10 admin-down admin-down ARPA 1514 25000000
TF0/0/1/11 admin-down admin-down ARPA 1514 25000000
TF0/0/1/12 admin-down admin-down ARPA 1514 25000000
TF0/0/1/13 admin-down admin-down ARPA 1514 25000000
TF0/0/1/14 admin-down admin-down ARPA 1514 25000000
TF0/0/1/15 admin-down admin-down ARPA 1514 25000000
Te0/0/1/24 admin-down admin-down ARPA 1514 10000000 /* slice 5 port numbers starts here */

Te0/0/1/25 admin-down admin-down ARPA 1514 10000000
Te0/0/1/26 admin-down admin-down ARPA 1514 10000000
Te0/0/1/27 admin-down admin-down ARPA 1514 10000000
Te0/0/1/28 admin-down admin-down ARPA 1514 10000000
Te0/0/1/29 admin-down admin-down ARPA 1514 10000000
Te0/0/1/30 admin-down admin-down ARPA 1514 10000000
Te0/0/1/31 admin-down admin-down ARPA 1514 10000000
Te0/0/1/32 admin-down admin-down ARPA 1514 10000000
Te0/0/1/33 admin-down admin-down ARPA 1514 10000000
Te0/0/1/34 admin-down admin-down ARPA 1514 10000000
Te0/0/1/35 admin-down admin-down ARPA 1514 10000000
Te0/0/1/36 admin-down admin-down ARPA 1514 10000000
Te0/0/1/37 admin-down admin-down ARPA 1514 10000000
Te0/0/1/38 admin-down admin-down ARPA 1514 10000000
Te0/0/1/39 admin-down admin-down ARPA 1514 10000000
Te0/0/1/40 admin-down admin-down ARPA 1514 10000000
Te0/0/1/41 admin-down admin-down ARPA 1514 10000000
Te0/0/1/42 admin-down admin-down ARPA 1514 10000000
Te0/0/1/43 admin-down admin-down ARPA 1514 10000000
```

```
Te0/0/1/44 admin-down admin-down ARPA 1514 10000000
Te0/0/1/45 admin-down admin-down ARPA 1514 10000000
Te0/0/1/46 admin-down admin-down ARPA 1514 10000000
Te0/0/1/47 admin-down admin-down ARPA 1514 10000000
```

This show command displays all 48 ports on Cisco A9903-8HG-PEC in 1GbE mode:

```
RP/0/RP0/CPU0:ios#show interfaces brief
-----------------------------------------------------------------
Fri Feb 24 10:33:06.517 UTC

Intf       Intf    LineP  Encap  MTU        BW
Name       State   State  Type (byte)    (Kbps)
------------------------------------------------------------------
Nu0           up   up     Null  1500          0
Mg0/RP0/CPU0/0up   up     ARPA  1514    1000000
Hu0/0/0/0    down  down   ARPA  1514  100000000
Hu0/0/0/1    down  down   ARPA  1514  100000000
Hu0/0/0/2    down  down   ARPA  1514  100000000
Hu0/0/0/3    down  down   ARPA  1514  100000000
Hu0/0/0/4    down  down   ARPA  1514  100000000
Hu0/0/0/5    up    up     ARPA  9216  100000000
Hu0/0/0/6    down  down   ARPA  1514  100000000
Hu0/0/0/7    down  down   ARPA  1514  100000000
Hu0/0/0/8    down  down   ARPA  1514  100000000
Hu0/0/0/9    down  down   ARPA  1514  100000000
Hu0/0/0/11   down  down   ARPA  1514  100000000
Hu0/0/0/12   down  down   ARPA  1514  100000000
Hu0/0/0/13   down  down   ARPA  1514  100000000
Hu0/0/0/15   up    up     ARPA  9216  100000000
Te0/0/0/16   down  down   ARPA  1514   10000000
Te0/0/0/17   down  down   ARPA  1514   10000000
Te0/0/0/18   down  down   ARPA  1514   10000000
Te0/0/0/19   down  down   ARPA  1514   10000000
Te0/0/0/20   down  down   ARPA  1514   10000000
Te0/0/0/21   down  down   ARPA  1514   10000000
Te0/0/0/22   down  down   ARPA  1514   10000000
Te0/0/0/23   down  down   ARPA  1514   10000000
Te0/0/0/24   down  down   ARPA  1514   10000000
Te0/0/0/25   down  down   ARPA  1514   10000000
Te0/0/0/26   down  down   ARPA  1514   10000000
Te0/0/0/27   down  down   ARPA  1514   10000000
Te0/0/0/28   down  down   ARPA  1514   10000000
Te0/0/0/29   down  down   ARPA  1514   10000000
Te0/0/0/30   down  down   ARPA  1514   10000000
Te0/0/0/31   down  down   ARPA  1514   10000000
Te0/0/0/32   down  down   ARPA  1514   10000000
Te0/0/0/33   down  down   ARPA  1514   10000000
Te0/0/0/34   down  down   ARPA  1514   10000000
Te0/0/0/35   down  down   ARPA  1514   10000000
Gi0/0/1/0    down  down   ARPA  1514    1000000/* slice 4 port numbers starts here  */
Gi0/0/1/1    down  down   ARPA  1514    1000000
Gi0/0/1/2    down  down   ARPA  1514    1000000
Gi0/0/1/3    down  down   ARPA  1514    1000000
Gi0/0/1/4    down  down   ARPA  1514    1000000
Gi0/0/1/5    down  down   ARPA  1514    1000000
Gi0/0/1/6    down  down   ARPA  1514    1000000
Gi0/0/1/7    down  down   ARPA  1514    1000000
Gi0/0/1/8    down  down   ARPA  1514    1000000
Gi0/0/1/9    down  down   ARPA  1514    1000000
Gi0/0/1/10   down  down   ARPA  1514    1000000
Gi0/0/1/11   down  down   ARPA  1514    1000000
Gi0/0/1/12   down  down   ARPA  1514    1000000
Gi0/0/1/13   up    up     ARPA  9216    1000000
Gi0/0/1/14   down  down   ARPA  1514    1000000
```

```
Gi0/0/1/15   up    up     ARPA  9216    1000000
Gi0/0/1/16   down  down   ARPA  1514    1000000
Gi0/0/1/17   down  down   ARPA  1514    1000000
Gi0/0/1/18   down  down   ARPA  1514    1000000
Gi0/0/1/19   down  down   ARPA  1514    1000000
Gi0/0/1/24   down  down   ARPA  1514    1000000/* slice 5 port numbers starts here  */
Gi0/0/1/25   down  down   ARPA  1514    1000000
Gi0/0/1/26   down  down   ARPA  1514    1000000
Gi0/0/1/27   down  down   ARPA  1514    1000000
Gi0/0/1/28   down  down   ARPA  1514    1000000
Gi0/0/1/29   down  down   ARPA  1514    1000000
Gi0/0/1/30   down  down   ARPA  1514    1000000
Gi0/0/1/31   down  down   ARPA  1514    1000000
Gi0/0/1/32   down  down   ARPA  1514    1000000
Gi0/0/1/33   down  down   ARPA  1514    1000000
Gi0/0/1/34   down  down   ARPA  1514    1000000
Gi0/0/1/35   down  down   ARPA  1514    1000000
Gi0/0/1/36   down  down   ARPA  1514    1000000
Gi0/0/1/37   down  down   ARPA  1514    1000000
Gi0/0/1/38   down  down   ARPA  1514    1000000
Gi0/0/1/39   down  down   ARPA  1514    1000000
Gi0/0/1/40   down  down   ARPA  1514    1000000
Gi0/0/1/41   down  down   ARPA  1514    1000000
Gi0/0/1/42   down  down   ARPA  1514    1000000
Gi0/0/1/43   down  down   ARPA  1514    1000000
Te0/0/1/20   down  down   ARPA  1514   10000000
Te0/0/1/21   down  down   ARPA  1514   10000000
Te0/0/1/22   down  down   ARPA  1514   10000000
Te0/0/1/23   down  down   ARPA  1514   10000000
Te0/0/1/44   down  down   ARPA  1514   10000000
Te0/0/1/45   down  down   ARPA  1514   10000000
Te0/0/1/46   down  down   ARPA  1514   10000000
Te0/0/1/47   down  down   ARPA  1514   10000000
```

Configuring Port Mode in Cisco ASR 9902 Router

The router has 48 ports with maximum of 800G data bandwidth capacity. You can configure the 48 ports in various port modes (100GbE, 25GbE, 10GbE, 40GbE and 1GbE) using the **hw-module location <node> slice <number> config-mode** command.

The port mode configuration is per slice and not per port. Also, the 48 ports on the Cisco ASR 9902 router are available in slice 0 and 1. Each slice can be configured for a mix of 1GbE, 10GbE, 25GbE, 40GbE or 100GbE port modes. The default configuration is 1x100GE, 1x100GE, 10x10GE, and 10x10GE.

For more information on the ports, see the Installing Modules and Cables in the Chassis chapter in the *Cisco ASR 9000 Series Fixed-Port Routers Hardware Installation Guide*.

The number of ports available on the router depends on the port mode configured on each slice. This table shows the configuration options and supported port modes on Cisco ASR 9902 router:

Table 22: Configuration Options, and Supported Port Modes on Cisco ASR 9902 Router

Configuration Options	Supported Port Modes on Slice 0 and Slice 1
Default	1x100GE, 1x100GE, 10x10GE, and 10x10GE
Option 1	1x100GE/1X40G/4X10GE, 1x100GE/1X40GE/4X10GE, 10x10GE or 5x1GE_5x10GE or 10x1GE, and 10x10GE or 5x1GE_5x10GE or 10x1GE

Configuration Options	Supported Port Modes on Slice 0 and Slice 1
Option 2	1x100GE/1X40GE/4X10GE, 1x100GE/1X40G/4X10GE, 4x25GE, and 10x10GE or 5x1GE_5x10GE
Option 3	1x100GE/1X40GE/4X10GE, 4x25GE, 4x25GE, and 1x100GE/1x40GE/4x10GE
Option 4	1x100GE/1X40GE/4X10GE, 1x100GE/1X40GE/4X10GE, 1x100GE/1X40GE, and 1x100GE/1X40GE/4X10GE

This sample shows how to configure Configuration Option 1:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 slice 0 config-mode
1x100GE,1x100GE,5x1GE_5x10GE,5x1GE_5x10GE
RP/0/RP0/CPU0:ios(config)#commit
```

This sample shows how to configure Configuration Option 1 with 10x1GE optics speed:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 slice 0 config-mode
1x100GE,1x100GE,10x1GE,10x1GE
RP/0/RP0/CPU0:ios(config)#commit
```

To revert to the default mode, use **no** form of the **hw-module location <node> slice <number> config-mode** command.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#no hw-module location 0/0/CPU0 slice 0 config-mode
1x100GE,1x100GE,5x1GE_5x10GE,5x1GE_5x10GE
RP/0/RP0/CPU0:ios(config)#commit
```

Use the **show interface brief** command to verify the ports configured using **config-mode** command.

Note The 40G port mode on the Cisco ASR 9902 router works only with Cisco IOS XR software version 7.8.x or higher. So, if you want to use the 40G port mode on the Cisco ASR 9902 router, you should update the Cisco IOS XR software version to 7.8.x or higher.

Configuring Port Modes in Cisco ASR 9903 Router and Cisco ASR 9000 Series Line Cards

Table 23: Feature History Table

Feature Name	Release Information	Description
1GbE Port Mode	Release 7.7.1	You can now use the 1GbE port mode option to configure 10GbE optics speed on the SFP+ ports. This allows you to enable SFP+ ports on the line cards or router to deliver 1GbE standard traffic by using the **hw-module location breakout** command. This option is enabled on the following Cisco ASR 9000 Series router and line cards: • ASR 9903 • A9K-4HG-FLEX-SE/TR • A99-4HG-FLEX-SE/TR • A9K-4HG-FLEX-FC • A99-4HG-FLEX-FC

The Cisco ASR 9903 router has 16 QSFP28 ports and 20 integrated SFP+ ports with a maximum of 1.6T data bandwidth capacity. You can configure the ports in various port modes (100GbE, 25GbE, 10GbE, and 1GbE) using the **hw-module location** *<node id>* **bay** *<bay number>* **port** *<port number>* **breakout** *<interface>* command.

For more information on the ports, see the Installing Modules and Cables in the Chassis chapter in the *Cisco ASR 9000 Series Fixed-Port Routers Hardware Installation Guide*.

This table shows the configuration options and supported port modes on Cisco ASR 9903 router and Cisco ASR 9000 Series line cards.

Table 24: Configuration Options and Supported Port Modes

Configuration Options	Supported Port Modes
Default	1x100GE, 1x100GE, 10x10GE, and 10x10GE
Option 1	1x100GE, 1x100GE, 10x10GE or 5x1GE_5x10GE or 10x1GE, and 10x10GE or 5x1GE_5x10GE or 10x1GE
Option 2	1x100GE, 1x100GE, 4x25GE, and 10x10GE or 5x1GE_5x10GE or 10x1GE
Option 3	1x100GE, 4x25GE, 4x25GE, and 1x100GE
Option 4	1x100GE, 1x100GE, 1x100GE, and 1x100GE

Configuring Supported Port Modes on Routers

To configure 1GbE port mode on Cisco ASR 9903 router, use the QSFP28 ports 10 and 14. You can configure the ports for a mix of 1GbE, 10GbE, 25GbE, or 100GbE port modes.

This sample shows how to configure port mode of 1GbE on ports 10 with 5x1GE-5x10GE option for Cisco ASR 9903 router:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 bay 0 port 10 breakout
5x1GE-5x10GE
RP/0/RP0/CPU0:ios(config)#commit
```

This sample shows how to configure port mode of 1GbE on ports 10 with 10x1GE option for Cisco ASR 9903 router:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 bay 0 port 10 breakout 10x1GE
RP/0/RP0/CPU0:ios(config)#commit
```

Configuring Supported Port Modes on Line Cards

To configure 1GbE port mode on the Cisco ASR 9000 Series line cards, use the **hw-module location** <*node*> **slice** <*number*> **config-mode** command. On these cards, the port mode configuration is done per slice and not per port.

For more information on the ports, see the Overview chapter in the *Cisco ASR 9000 Series Aggregation Services Router Ethernet Line Card Installation Guide*.

This sample shows how to configure Option 1 on slice 0 with 1x100GE, 1x100GE, 5x1GE_5x10GE, and 5x1GE_5x10GE port modes on the line cards:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 slice 0 config-mode
1x100GE,1x100GE,5x1GE_5x10GE,5x1GE_5x10GE
RP/0/RP0/CPU0:ios(config)#commit
```

This sample shows how to configure Option 1 on slice 0 with 1x100GE, 1x100GE, 10x10GE, and 10x1GE port modes on the line cards:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/CPU0 slice 0 config-mode
1x100GE,1x100GE,10x1GE,10x1GE
RP/0/RP0/CPU0:ios(config)#commit
```

To revert to the default mode, use the **no** form of the **hw-module location** <*node*> **slice** <*number*> config-mode command.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#no hw-module location 0/0/CPU0 slice 0 config-mode
1x100GE,1x100GE,5x1GE_5x10GE,5x1GE_5x10GE
RP/0/RP0/CPU0:ios(config)#commit
```

Configure Single Feed Power Mode

Cisco ASR 9000 series router supports the operating of one or all power modules. For example, V1 DC, V2 DC, V3 AC and V3 DC.

Ideally, you're expected to connect all the power modules (or feed) to power supply. If you don't connect any one feed, the system raises an alarm or error message.

You can configure the single-feed power mode to suppress the error message or an alarm for any missing feeds.

Configuration Example

The following example enables the single power feed mode for the 0/PS2/M0/SP power module:

```
Router#admin
Router(admin)#config
Router(admin-config)#power single-feed location 0/PS2/M0/SP
```

Excluding Sensitive Information in Show Running Configurations Output

Table 25: Feature History Table

Feature Name	Release Information	Feature Description
Excluding Sensitive Information in Show Running Configurations Command Output	Release 7.5.4	You can now exclude sensitive information such as strings, usernames, passwords, comments, or IP addresses within the **show running-configuration** command output by enabling sanitization on the nonvolatile generation (NVGEN) process. With this feature, you can achieve better data protection to prevent cybersecurity risks compared to regular router algorithms. This feature introduces the **nvgen default-sanitize** command.

The **show running configuration** command uses the nonvolatile generation (NVGEN) process in IOS-XR software to collect configuration information from every system component and construct a running configuration file to create its output. However, this file may contain sensitive information, including usernames, passwords, and IP addresses, which could pose a security threat when obfuscation algorithms in the router are weak compared to modern cryptographic standards.

In this feature, you can mask the following types of sensitive information in the show running configurations:

- Strings
- Usernames
- Passwords
- Comments
- IP Addresses

On enabling the sanitization in show running configurations, the NVGEN process replaces the corresponding information with **<removed>** string. For example, if you enable sanitization for IP Addresses, the show running configuration includes the **<removed>** string in place of all the IP Addresses in the output.

Sanitizing Strings

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize strings
Router:(config)# commit
```

Running Configuration

```
Router# show run nvgen
nvgen
 default-sanitize strings
!
```

Verification

```
Router# show run int Hu0/2/0/4
interface HundredGigE0/2/0/4
 ! This is comment 1
 description <removed>
 !
```

Sanitizing Usernames

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize usernames
Router:(config)# commit
```

Running Configuration

```
Router# show run nvgen
nvgen
 default-sanitize usernames
!
```

Verification

```
Router# show run username test
username <removed>
 group root-lr
 password 7 172864HJWBJHBCWH
!
```

Sanitizing Passwords

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize passwords
Router:(config)# commit
```

Running Configuration

```
Router# show run nvgen
nvgen
 default-sanitize passwords
!
```

Verification

```
Router# show run username test
username test
 group root-lr
 password 7 <removed>
!
```

Sanitizing Comments

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize comments
Router:(config)# commit
```

Running Configuration

```
Router# show run nvgen
nvgen
 default-sanitize comments
!
```

Verification

```
Router# show run int Hu0/2/0/4
interface HundredGigE0/2/0/4
 ! <comments removed>
 description This is bundle member
 !
```

Sanitizing IP Addresses

Configuration

```
Router# config
Router:(config)# nvgen default-sanitize ipaddrs
Router:(config)# commit
```

Verification

```
Router# show run int Hu0/2/0/4
interface HundredGigE0/2/0/4
 ! This is comment 1
 description This is bundle member
  ipv4 address <removed> <removed>
!
```

Additional References

The following sections provide references related to hardware management on Cisco IOS XR software.

Related Documents

Related Topic	Document Title
Cisco IOS XR hardware commands	Hardware Redundancy and Node Administration Commands on *the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*

Related Topic	Document Title
Cisco IOS XR hardware documentation	See Cisco Carrier Routing System Install and Upgrade Guides at: http://www.cisco.com/en/US/products/ps5763/prod_installation_guides_list.html
Information about getting started with Cisco IOS XR software	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
ROM Monitor	*ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*
Cisco IOS XR command master list	*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*
Information about user groups and task IDs	*Configuring AAA Services on the Cisco ASR 9000 Series Router* module of *System Security Configuration Guide for Cisco ASR 9000 Series Routers*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 8

Upgrading and Managing Cisco IOS XR Software

Cisco IOS XR software is divided into software packages so that you can select which features run on your router. This module describes the concepts and tasks necessary to add feature packages, upgrade the active set of packages, roll back to a previously active set of packages, and perform other related package management tasks.

For complete descriptions of the commands listed in this module, see Related Documents, on page 224. To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 26: Feature History for Upgrading and Managing Cisco IOS XR Software

Release	Modification
Release 3.7.2	The feature was introduced.
Release 3.9.0	No modification.
Release 4.0.0	A procedure to upgrade software from Cisco IOS XR Release 3.x was introduced. See Upgrading to Cisco IOS XR Software Release 4.0, on page 208. Support for installation commands was removed from EXEC mode. The ability to install software on a specific SDR was removed.
Release 6.0.1	Support was added for digitally signed images. See Digitally Signed Images , on page 175
Release 6.3.1	Support for parallel FPD upgrade for power modules was added.

This module contains the following topics:

Overview of Cisco IOS XR Software Packages

Cisco IOS XR software is divided into software packages so that you can select which features run on your router. Each package contains the components to perform a specific set of router functions, such as routing, security, or modular services card (MSC) support. Bundles are groups of packages that can be downloaded as a set. For example, Cisco IOS XR Unicast Routing Core Bundle (known as *mini*) provides the main packages for use on every router.

Adding a package to the router does not affect the operation of the router—it only copies the package files to a local storage device on the router, known as the *boot device* (such as the compact flash drive). To make the package functional on the router, you must activate it for one or more cards.

To upgrade a package, you activate a newer version of the package. When the automatic compatibility checks have been passed, the new version is activated, and the old version is deactivated.

Note Activating a software maintenance upgrade (SMU) does not cause any earlier SMUs or the package to which the SMU applies to be automatically deactivated.

Note If an interface on a router does not have a configuration and is brought up by performing no-shut operation, then upon router reload, the interface state changes to **admin-shutdown** automatically.

To downgrade a package, you activate an older version of the package. When the automatic compatibility checks have been passed, the older version is activated, and the newer version is deactivated.

Caution Do not perform any install operations when the router is reloading.

Note For more information on the features and components included in each package, refer to the release notes.

Package Installation Envelopes

Package Installation Envelopes (PIEs) are nonbootable files that contain a single package or a set of packages (called a *composite package* or *bundle*). Because the files are nonbootable, they are used to add software package files to a running router.

PIE files have a `pie` extension. When a PIE file contains software for a specific bug fix, it is called a *software maintenance upgrade* (SMU).

Note Files with the `vm` extension are bootable installation files used only to replace all current Cisco IOS XR software. These files are installed from ROM Monitor mode, which causes significant router downtime. Cisco Systems recommends installing or upgrading software packages only using PIE files as described in this document. For more information on `vm` files, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.

Summary of Cisco IOS XR Software Packages

Every router includes a basic set of required packages contained in the Cisco IOS XR Unicast Routing Core Bundle. Additional optional packages can be added and activated on the router to provide specific features.

Packages in the Cisco IOS XR Unicast Routing Core Bundle

The packages contained in the Cisco IOS XR Unicast Routing Core Bundle are as follows:

- Operating system (OS) and minimum boot image (MBI)—Kernel, file system, memory management, and other slow changing core components.
- Base—Interface manager, system database, checkpoint services, configuration management, other slow-changing components.
- Infra—Resource management: rack, fabric.
- Routing—RIB, BGP, ISIS, OSPF, EIGRP, RIP, RPL, and other routing protocols.
- Forwarding—FIB, ARP, QoS, ACL, and other components.
- LC— Line card drivers.

The filename for this bundle is: `asr9k-mini.pie-`*`version`*.

Refer to the release notes for additional information on the specific features provided by each package.

Software Maintenance Upgrades

A software maintenance upgrade (SMU) is a PIE file that contains fixes for a specific defect. A composite SMU is a PIE file that contains SMUs for more than one package. SMUs are added and activated using the same procedures as other PIE files. SMUs are created to respond to immediate issues and do not include new features. Typically, SMUs do not have a large impact on router operations. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.

The affect of an SMU depends on its type:

- Process Restart SMU—Causes a process or group of processes to restart on activation.
- Reload SMU—Causes a parallel reload (of RPs and line cards).

SMUs are not an alternative to maintenance releases. They provide quick resolution of immediate issues. All bugs fixed by SMUs are integrated into the maintenance releases. For information on available SMUs, contact Cisco Technical Support, as described in *Obtaining Technical Assistance* in the monthly What's New in Cisco Product Documentation.

Note Activating a software maintenance upgrade (SMU) does not cause any earlier SMUs, or the package to which the SMU applies, to be automatically deactivated.

Third-party SMUs

Consider these points while activating and deactivating third-party SMUs:

- To activate a third-party SMU you should have a corresponding base package.
- When you activate a third-party SMU, the corresponding third-party base package state is inactive, this is an expected behavior.
- To deactivate a third-party SMU, you should activate corresponding third-party base package.

Related Topics

Updating Software Images Without a Router Reload

PIE Filenames and Version Numbers

PIE filenames have two formats: one for composite-package PIEs (bundles) and one for single-package PIEs. A *composite-package file* is a PIE file that contains multiple packages.

Note Hyphens in the filename are part of the filename.

Table 27: PIE Filenames, on page 170 shows the filenames for available PIE types.

Table 27: PIE Filenames

Software Delivery Type	Filename	Example
Composite (Bundle) PIE	*platform-composite_name*.pie-*major.minor.maintenance*	`asr9k-mini.pie-3.7.2`
Single package PIE	*platform-package_type*.-p.pie-*major.minor.maintenance*	`asr9k-mpls.pie-3.7.2`
Composite SMU	comp-*platform-composite_name.ddts*.pie	`comp-asr9k-001.CSCec98xxx.pie`
Single package SMU	*platform-package_type-major.minor.maintenance.ddts*.pie	`asr9k-base-3.7.2.CSCei45xxx.pie`
Note A SMU composite name usually is “001”, which means the SMU is the first SMU for that DDTS. In rare cases in which the same DDTS requires multiple composite SMUs, a second composite version number is released as “002”. In the previous example, a second composite SMU “comp-002.CSCec98766” would be created for DDTS CSCec98766.		

Filename Component Description

The filename components for all packages are described in Table 28: Composite- and Single-Package Filename Components, on page 171.

Table 28: Composite- and Single-Package Filename Components

Component	Description
platform	Identifies the platform for which the software package is designed. • The platform designation is "asr9k."
composite_name	Identifies a specific composite package. • The only composite PIE file at this time is named "mini" and includes all packages described in the Cisco IOS XR Unicast Routing Core Bundle.
package_type	Identifies the type of package the file supports (*package_type* applies only to single-package PIEs). Package types include: • mcast—Multicast package • mgbl—Manageability package • mpls—MPLS package • k9sec—Security package • diags—Diagnostics package • fpd—Field-programmable device package • doc—Documentation package
major	Identifies the major release of this package. • A major release occurs when there is a major architectural change to the product (for example, a major new capability is introduced). • All packages operating on the router must be at the same major release level. • A major release is the least frequent release and may require a router reboot.

minor	Identifies the minor release of this package.
	• A minor release contains one or more of the following:
	  • New features
	  • Bug fixes
	• The minor release version does not have to be identical for all software packages operating on the router, but the operating packages must be certified by Cisco as compatible with each other.
	• A minor release may require a router reboot.

Component	Description
maintenance	Identifies the maintenance release of this package. • A maintenance release contains a collection of bug fixes for a package. • The maintenance release version does not have to be identical for all software packages operating on the router, but the major and minor versions of the maintenance release must match those of the package being updated. • A maintenance release does not usually require a router reboot.
ddts	SMUs only. Identifies a DDTS[8] number that describes the problem this SMU addresses. DDTS is the method used to track known bugs and the resolutions or workarounds for those issues.
px	Identifies images that are compatible with hardware that uses the x86 architecture. Starting with Cisco IOS XR Release 4.2, -px releases replace the -p releases.

8 distributed defect tracking system

Copying the PIE File to a Local Storage Device or Network Server

To add an optional package or upgrade or downgrade a package, you must copy the appropriate PIE file to a local storage device or to a network file server to which the router has access.

If you need to store PIE files on the router, we recommended storing PIE files on the hard disk. Flash disk0: serves as the boot device for packages that have been added or activated on the system. Flash disk1: is used as a backup for disk0:.

Tip Before copying PIE files to a local storage device, use the **dir** command to check to see if the required PIE files are already on the device.

Information About Package Management

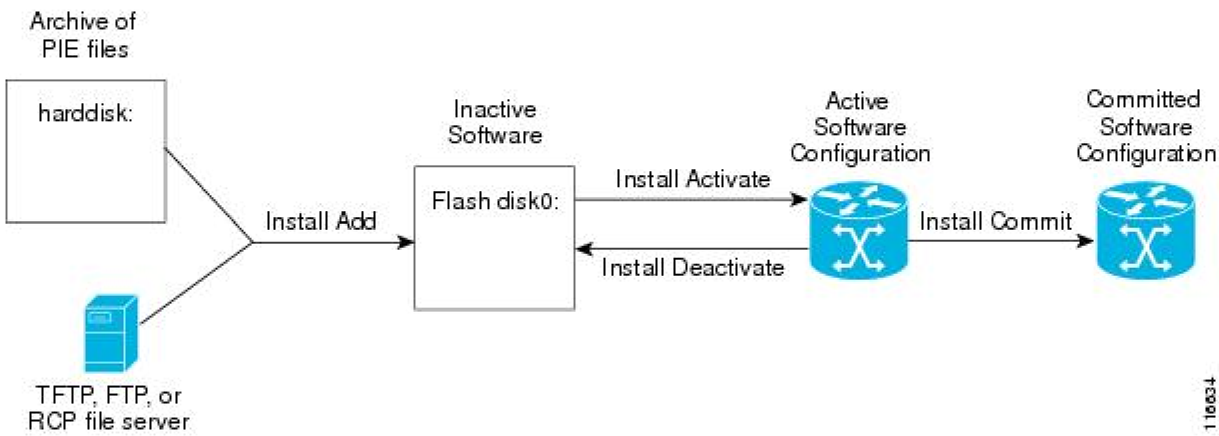
Summary of Package Management

The general procedure for adding optional packages, upgrading a package or package set, or downgrading packages on the router is as follows:

1. Copy the package file or files to a local storage device or file server.
2. Add the package or packages on the router using the command **install add**.
3. Activate the package or packages on the router using the **install activate** command.
4. Commit the current set of packages using the **install commit** command.

Figure 8: Process to Add, Activate, and Commit Cisco IOS XR Software Packages, on page 173 illustrates key steps in the package management process.

Figure 8: Process to Add, Activate, and Commit Cisco IOS XR Software Packages



Adding Packages

Use the **install add** command to unpack the package software files from a PIE file and copy them to the boot device (usually disk0:) of your router.

From administration EXEC mode, the package software files are added to the boot device of the DSC of the router, as well as all active and standby Route Processors (RPs) and fabric shelf controllers (SCs) installed on the router.

Note The disk that holds the unpacked software files is also known as the *boot device*. By default, flash disk0: is used as the boot device. To use an alternate storage device, such as flash disk1:, see the *Router Recovery with ROM Monitor* module of *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*. Remember that all RSPs in a system must use the same boot device. If the boot device on the primary RSP is flash disk0:, then the standby RSP must also have a flash disk0:.

Verifying Package Details

Before you activate a package on the router, you can verify the type of upgrade that is required for the package and whether the package requires a router reload or not. Use the **show install package** *pie* **detail** command in admin mode.

```
RP/0/RSP0/CPU0:router(admin)# show install package disk0:asr9k-px-4.x.x.04I.CSCuc66088-0.0.4.i
 detail

Mon Nov 19 09:44:24.036 UTC
disk0:asr9k-px-4.x.x.04I.CSCuc66088-0.0.4.i
    asr9k-px-4.x.x.04I.CSCuc66088 V0.0.4.i[SMU]  User specified bundle
        iosxr-infra-asr9k-px1-4.x.x.04I.CSCuc66088.pi.pie.
     [composite package]
     [root package, grouped contents]
    Vendor : Cisco Systems
    Desc   : User specified bundle iosxr-infra-asr9k-px1-4.x.x.04I.CSCuc66088.pi.pie.
    Build  : Built on Fri Nov  9 11:00:11 UTC 2012
    Source : By iox-bld27 in /scratch1/SMU_BLD_WS/ci-431_206626_CSCuc66088_121109102249 for
 pie
    Card(s): RP, CRS-RP-X86, CRS8-RP-x86, CRS16-RP-x86, ASR9001-RP, RP-STARSCREAM,
```

```
NP24-4x10GE,
           NP24-40x1GE, NP40-40x1GE, NP40-4x10GE, NP40-8x10GE, NP40-2_20_COMBO, NP80-8x10GE,

             NP80-16x10GE, NP200-24x10GE, NP200-36x10GE, NP200-2x100GE, NP200-1x100GE,
NP200-5x40GE,
             NP200-8x10GE, NP200-MOD-SMEM, NP200-MOD-LMEM, ASR9001-LC, A9K-SIP-700,
A9K-SIP-500, A9K-SIP-AVSM
    Restart information:
      Default:
        parallel impacted processes restart
    Size Compressed/Uncompressed: 1744KB/1830KB (95%)
    Components in package disk0:asr9k-px-4.x.x.04I.CSCuc66088-0.0.4.i, package
asr9k-px-4.x.x.04I.CSCuc66088:
    disk0:iosxr-infra-4.x.x.04I.CSCuc66088-0.0.4.i
        iosxr-infra-4.x.x.04I.CSCuc66088 V0.0.4.i[SMU]  IOS-XR Infra Package Definition
        Vendor : Cisco Systems
        Desc   : IOS-XR Infra Package Definition
        Build  : Built on Fri Nov  9 11:00:10 UTC 2012
       Source : By iox-bld27 in /scratch1/SMU_BLD_WS/ci-431_206626_CSCuc66088_121109102249
 for pie
        Card(s): RP, CRS-RP-X86, CRS8-RP-x86, CRS16-RP-x86, ASR9001-RP, RP-STARSCREAM,
NP24-4x10GE,
                 NP24-40x1GE, NP40-40x1GE, NP40-4x10GE, NP40-8x10GE, NP40-2_20_COMBO,
NP80-8x10GE,
                 NP80-16x10GE, NP200-24x10GE, NP200-36x10GE, NP200-2x100GE, NP200-1x100GE,

                 NP200-5x40GE, NP200-8x10GE, NP200-MOD-SMEM, NP200-MOD-LMEM, ASR9001-LC,
                 A9K-SIP-700, A9K-SIP-500, A9K-SIP-AVSM

        Size Compressed/Uncompressed: 1744KB/1830KB (95%)
        Components in package disk0:iosxr-infra-4.x.x.04I.CSCuc66088-0.0.4.i,
            package  iosxr-infra-4.x.x.04I.CSCuc66088:
            platforms-spa-chopper V[ci-4x-bugfix/8]  This component contains Platform
Independent
            Chopper SPA Code.
            iosxr-infra-4.x.x.04I.CSCuc66088-package V[Default]  Manifest information for
package
            iosxr-infra-4.x.x.04I.CSCuc66088
            iosxr-infra-4.x.x.04I.CSCuc66088-package-compatibility V[Default]
            Package Compatibility information for package iosxr-infra-4.x.x.04I.CSCuc66088
```

Activating Packages

Software packages remain inactive until activated with the **install activate** command.

After a package has been added to the router , use the **install activate** command to activate the package or SMUs for all valid cards. Information within the package is used to verify compatibility with the target cards and with the other active software. Actual activation is performed only after the package compatibility and application programming interface (API) compatibility checks have been passed.

Activating a Package on the Router

To activate a package on your router, use the **install activate** command in administration EXEC mode. The **install activate** command also activates the package on all administration plane nodes and resources, including service processors (SPs), fabric SCs, fan controllers, alarm modules, and power modules.

Activating Multiple Packages or SMUs

To install multiple packages or software maintenance upgrades (SMUs) with a single command, use the **install activate** command and either specify up to 16 packages by repeating *device: package* arguments or use wildcard syntax to specify multiple packages. Some SMUs may require a reload. If the operation requires a node reload, the user is prompted before the installation operation occurs.

> **Note** After activating SMU CSCwc03813, ensure that you either reload the Line Card or remove and reapply the existing Access Control Lists, for the updates to take effect.

Related Topics

SMU Installation Combinations

Activating All Packages Added in a Specific Operation

To install all packages that were added in a specific **install add** operation, use the **install activate** command with the **id** *add-id* keyword and argument, specifying the operation ID of the **install add** operation. You can specify up to 16 operations in a single command.

Adding and Activating a Package with a Single Command

To add and activate a package with a single command, use the **install add** command with the **activate** keyword from administration EXEC mode.

Digitally Signed Images

From Release 6.0.1 onwards, all Cisco IOS XR images are digitally signed to ensure the authenticity of the software. If the ROMMON/BIOS image on the line card is digitally signed, then ensure that the Cisco IOS XR image on the router is also digitally signed, as the ROMMON/BIOS checks the Cisco IOS XR image for a digital signature.

> **Note** The Cisco IOS XR Release 5.3.x images are not digitally signed. If you are running Cisco IOS XR Release 5.3.x, then ensure that you install the Abraxas based SMU for successful operation of the router with the latest line cards.

The digitally signed images are supported on the following line cards:

- A99-RP2-TR
- A99-RP2-SE
- A9K-RSP880-TR
- A9K-RSP880-SE
- A9K-8X100GE-SE
- A9K-8X100GE-TR
- A9K-4X100GE-SE
- A9K-4X100GE-TR
- A99-8X100GE-SE
- A99-8X100GE-TR

- A9K-MOD400-SE
- A9K-MOD400-TR
- A9K-MOD200-SE
- A9K-MOD200-TR
- A9K-400G-DWDM-TR
- A99-12X100GE

For more information on installing line cards, see *Cisco ASR 9000 Series Aggregation Services Routers Ethernet Line Card Installation Guide*.

Validation Messages

When image credentials, signature, and other attributes of the signature envelope are matched, the router reboots.

The following message is displayed on successful image validation.

```
File reception completed.**** check image validation ****.......BIOS CODE SIGN ENTRY ...Image
 ASR9K-Tomahawk verified successfully~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

If image validation is unsuccessful, the boot process is interrupted, and the router enters the ROMMON CLI mode.

The following message is displayed on unsuccessful image validation.

```
**** check image validation ****.......BIOS CODE SIGN ENTRY ... Image ASR9K-Tomahawk
Verification Failed Invalid signature offset *********************************** ASR9K
image validation failed, cannot boot image, contact tech support
***********************************
```

Upgrading and Downgrading Packages

To upgrade a package, activate the latest version of the package; the previous version is automatically deactivated. To downgrade a package, activate the previous version of the package; the latest version is automatically deactivated.

Actual activation is performed only after compatibility checks have been passed.

Note

- Activating a software maintenance upgrade (SMU) does not cause previous versions of the SMUs, or the package to which the SMU applies, to be automatically deactivated.
- If you upgrade an ASR 9000 router with low RSP card memory, then the RSP440-TR route-switch processor and Cisco ASR 9000 2nd Generation line card can become inaccessible due to insufficient memory. Power cycling the router may help bring the router back to the up state.

Committing the Active Software Set

When a package is activated on the router , it becomes part of the current running configuration. To make the package activation persistent across reloads, enter the **install commit** command in administration EXEC mode. On startup, the designated shelf controller (DSC) of the secure domain router (SDR) loads the committed software set.

> **Note** If the system is restarted before the active software set is saved with the **install commit** command, the previously committed software set is used.

Rolling Back to a Previous Installation Operation

Although the term *commit* sounds final, the Cisco IOS XR software provides the flexibility to roll back the selected package set to previously saved package sets. Each time a package is activated or deactivated, a rollback point is created that defines the package set that is active after the package activation or deactivation. The software also creates a rollback point for the last committed package set. If you find that you prefer a previous package set over the currently active package set, you can use the **install rollback** command to make a previously active package set active again.

Related Topics

Rolling Back to a Previous Software Set, on page 220

Multiple Disks Support during Installations

In installations on platforms where Cisco IOS XR Software is supported, only a single disk is used as an install device; that is, either disk0 or disk1. When multiple packages are installed on a single disk, it results in space constraints. To resolve this space limitation, the disk supported for the install operations has been extended to another disk called the disk1. When installing multiple packages, this feature enables you to choose between disk0 and disk1.

To add packages to a specific disk name, use the **install media** command in the admin configuration mode.

```
RP/0/RSP0/CPU0: router (admin) # install media disk1
```

Restrictions

- Before enabling the addition of disk1 through the **install media** command, the disk mirroring feature should be explicitly disabled. For details regarding disk mirroring, see the Disk Mirroring chapter.
- All single version packages should be installed into one disk; that is, either disk0 or disk1.
- When downgrading to an image that does not support extended disk, the rollback points of the extended disk will not be available on the downgraded image. For example, assume a case where the version1 (V1) image does not support the extended disk functionality and version2 (V2) image supports the functionality. Upgrading from V1(disk0) to V2(disk1), in such a case, makes the rollback points of V1 available on V2. However, when downgrading from V2(disk1) to V1(disk0), the rollback points of V2 will not be available on V1. For more information about the rollback feature and rollback points, see the Upgrading and Managing Software chapter.

Deactivation of fully superseded SMUs

Cisco IOS XR Software will accumulate a set of Software Maintenance Upgrades (SMUs) over time, where an older SMU gets superceded by the latest SMU. For example, if SMU A was initially delivered to you, and subsequently, as a result of a bug resolution, SMU B was delivered, then SMU A becomes the subset of SMU B and SMU A is superceded by SMU B. In this case, SMU A is redundant and can be deactivated to clean up the software package.

> **Note** When an older SMU gets superceded their code is no longer used but it can be used for rollback purposes which consumes the disk space.

When you install a reload SMU which supercedes the existing SMU the router reboots and SMU is placed in the superceded list automatically. If the superceded reload SMU is deactivated and fully supercedes the existing SMU, then router will not reboot when you run the **install deactivate superceded** command and the **install remove inactive** command.

> **Note** Removing the superceded SMUs will increase the disk space and it will not affect the functionality.

To deactivate all the fully superseded SMUs, use the **install deactivate superseded** command in the admin mode.

```
RP/0/RSP0/CPU0: router(admin) # install deactivate superseded
```

To display the details of the SMUs that are superseded, use the **show install superseded** command in the EXEC mode.

```
RP/0/RSP0/CPU0: router # show install superseded
Thu Feb 3 17:37:20.379 UTC
disk0:asr9k-px-4.3.0.CSCud93518-1.0.0 is fully superseded by
disk0:asr9k-px-4.3.0.CSCue23747-1.0.0
```

Support for the Ignore Package Presence Check Option

During any software package upgrade in Cisco IOS XR Software, two versions of the packages get stored, both the previous version and the upgraded version. In Route Switch Processor 2 (RSP2), the disk space is insufficient to hold all packages of these two versions. To address this, a new optional keyword, **ignore-pkg-presence-check**, is added to the **install activate** command, which allows upgrading with lesser number of packages. For example, assume a case where version1 (V1) of the software consists of packages A, B, C, and D, and you want to upgrade to the version2 (V2) with only 3 packages (A, B, and C). The ignore-pkg-presence-check option allows only packages A, B, and C to be upgraded to V2 and deactivates package D of V1. Thus, an explicit deactivation of package D is not required and the user can add package D of V1 after upgrading to V2.

To upgrade software with lesser number of packages, use the **install activate [ignore-pkg-presence-check]** command in the admin mode.

```
RP/0/RSP0/CPU0: router(admin) # install activate [ignore-pkg-presence-check] V2 packages
```

Restrictions

The restrictions for this option are:

- The ignore-pkg-presence-check keyword is supported only with the **install activate** command and is not supported with the **install add activate** command.
- When you upgrade using the ignore-pkg-presence-check option, the deactivation of packages always happens synchronously, using the synchronous keyword in the **install deactivate** command.

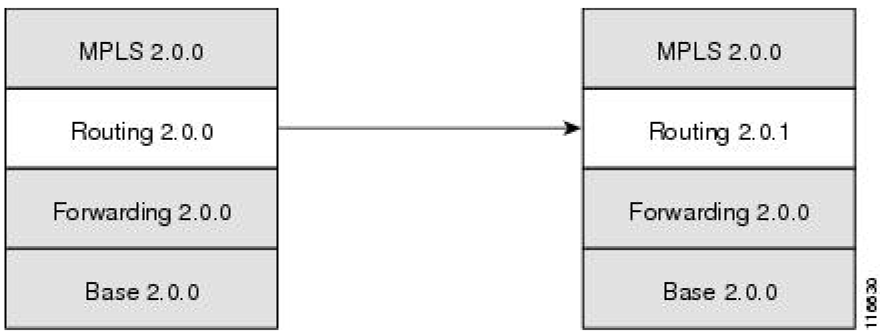
Upgrading Packages

To upgrade a package that is currently active on your router, add and activate a newer version of the same package (see Figure 9: Example of a Maintenance Release Package Upgrade, on page 179). The older version of the software package is deactivated automatically. These actions are permitted only after the package compatibility checks and API version compatibility checks have been passed.

Deactivated packages are not removed from the router. To remove inactive package files, use the **install remove** command.

Caution Upgrading or downgrading a software package can cause a process to restart or a new process to start. Use the **test** option to preview the impact of the package activation.

Figure 9: Example of a Maintenance Release Package Upgrade



Related Topics

Deactivating and Removing Cisco IOS XR Software Packages, on page 215

Downgrading Packages

To downgrade a software package, activate an older version on one or more cards for which that package is already active. The newer version of the same software package is deactivated automatically. These actions are performed only after the package compatibility checks and API version compatibility checks have been passed.

Deactivated packages are not removed from the router. To remove inactive package files, use the **install remove** command. See the *Related Topics* section for links to more information.

Note

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

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

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

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

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

Related Topics

Deactivating and Removing Cisco IOS XR Software Packages, on page 215

Impact of Package Version Changes

Each package version change has a different impact on the operation of the router, depending on the type of package and whether the upgrade is for a major, minor, or maintenance release. The following resources can provide more information on the impact of a package version change:

- See *Related Topics* for more information on the typical impact for major, minor, and maintenance releases.
- For specific information about the impact of an upgrade, consult the release notes for the package release, and test the impact of the package activation by adding the test option to the **install activate** command.
- The Cisco IOS XR Software Selector tool also contains information on package version compatibility.

Related Topics

PIE Filenames and Version Numbers, on page 170
Obtaining and Placing Cisco IOS XR Software, on page 185

Impact of Package Activation and Deactivation

Activation or deactivation of a package can have an immediate impact on the system. The system can be affected in the following ways:

- When a new package is activated, any new CLI commands for the package are added to the router. The router need not be restarted or reloaded.
- When a package is deactivated, the commands associated with the features being deactivated are removed from the router. The commands are no longer available to the user.
- During a software package deactivation, upgrade, or downgrade, any incompatible configurations are removed from the running configuration of the router, and saved to a file. Messages for incompatible configurations are displayed. Incompatible configurations are those configurations that are not supported by the new version of the software package.

Note

You must address any issues that result from the revised configuration and reapply the configuration, if necessary.

- New processes may be started.
- Running processes may be stopped or restarted.
- All processes in the cards may be restarted. Restarting processes in the cards is equivalent to a soft reset.
- The cards may reload.
- No impact: no processes in the card may be affected.

Tip When activating and deactivating packages, use the **test** option to test the effects of a command without impacting the running system. After the activation or deactivation process completes, enter the **show install log** command to display the process results.

Delaying the Return of the CLI Prompt

By default, the CLI prompt is returned to the screen before the installation operation is complete, which allows you to enter other commands that are not installation commands. If additional installation requests are attempted before the first operation is complete, they are not run.

To delay the return of the CLI prompt until an installation operation is complete, enter the **install** command with the **synchronous** keyword. For example:

```
install add disk1:/pie-file synchronous
install activate disk0:package synchronous
```

To determine if an **install** command is currently running, enter the **show install request** command.

Displaying Installation Log Information

The install log provides information on the history of the installation operations. Each time an installation operation is run, a number is assigned to that operation.

- Use the **show install log** command to display information about both successful and failed installation operations.
- The **show install log** command with no arguments displays a summary of all installation operations. Specify the *request-id* argument to display information specific to an operation. Use the **detail** or **verbose** keywords to display details for specific operation.
- Use the **detail** or **verbose** keywords to display detailed information, including file changes, nodes that could be reloaded, impact to processes, and impact to Dynamic Link Libraries (DLLs).

Tip By default, the install log stores up to 50 entries. Use the **clear install log-history** command to reset the number of entries to any value from 0 to 255.

Examples

Displaying install log Entries: Example

The following example displays information for the install requests. Use the **verbose** keyword to display detailed information, including files changes, impact to processes, and impact to DLLs.

```
RP/0/RSP0/CPU0:router(admin)# show install log verbose

Install operation 1 started by user 'labuser' at 17:48:51 UTC Sat Jun 03 2009.
install add /disk1:asr9k-diags-p.pie-PD34-06.06.07
/disk1:asr9k-k9sec-p.pie-PD34-06.06.07 /disk1:asr9k-mcast-p.pie-PD34-06.06.07
/disk1:asr9k-mgbl-p.pie-PD34-06.06.07 /disk1:asr9k-mpls-p.pie-PD34-06.06.07
Install operation 1 completed successfully at 17:51:32 UTC Sat Jun 03 2009.

Install logs:
 Install operation 1 'install add /disk1:asr9k-diags-p.pie-PD34-06.06.07
 /disk1:asr9k-k9sec-p.pie-PD34-06.06.07 /disk1:asr9k-mcast-p.pie-PD34-06.06.07
 /disk1:asr9k-mgbl-p.pie-PD34-06.06.07 /disk1:asr9k-mpls-p.pie-PD34-06.06.07'
 started by user 'labuser' at 17:48:51 UTC Sat Jun 03 2009.
 Info:     The following packages are now available to be activated:
    Info:
    Info:         disk0:asr9k-diags-3.7.2.1I
    Info:         disk0:asr9k-k9sec-3.7.2.1I
    Info:         disk0:asr9k-mcast-3.7.2.1I
    Info:         disk0:asr9k-mgbl-3.7.2.1I
    Info:         disk0:asr9k-mpls-3.7.2.1I
    Info:
  Install operation 1 completed successfully at 17:51:32 UTC Sat Jun 03 2009.
  Install operation 2 started by user 'labuser' at 18:06:32 UTC Sat Jun 03 2009.
  install activate disk0:asr9k-diags-3.7.2.1I disk0:asr94k-k9sec-3.7.2.1I
  disk0:asr9k-mcast-3.7.2.1I disk0:asr9k-mgbl-3.7.2.1I disk0:asr9k-mpls-3.7.2.1I
  Install operation 2 completed successfully at 18:07:48 UTC Sat Jun 03 2009.
    Summary:
      Install method: parallel
      Summary of changes on nodes 0/1/SP, 0/6/SP, 0/SM0/SP, 0/SM1/SP,
      0/SM2/SP,0/SM3/SP:
          Activated:    asr9k-diags-3.7.2.1I
          No processes affected

      Summary of changes on nodes 0/1/CPU0, 0/6/CPU0:
          Activated:    asr9k-diags-3.7.2.1I
                        asr9k-mcast-3.7.2.1I
                        asr9k-mpls-3.7.2.1I
         1 asr9k-mpls processes affected (0 updated, 1 added, 0 removed, 0 impacted)
         2 asr9k-mcast processes affected (0 updated, 2 added, 0 removed, 0 impacted)

      Summary of changes on nodes 0/RP0/CPU0, 0/RP1/CPU0:
          Activated:    asr9k-diags-3.7.2.1I
                        asr9k-k9sec-3.7.2.1I
                        asr9k-mcast-3.7.2.1I
                        asr9k-mgbl-3.7.2.1I
                        asr9k-mpls-3.7.2.1I
         6 asr9k-mgbl processes affected (0 updated, 6 added, 0 removed, 0 impacted)
         8 asr9k-mpls processes affected (0 updated, 8 added, 0 removed, 0 impacted)
         7 asr9k-k9sec processes affected (0 updated, 7 added, 0 removed, 0 impacted)
        14 asr9k-mcast processes affected (0 updated, 14 added, 0 removed, 0 impacted)

  Install logs:
    Install operation 2 'install activate disk0:asr9k-diags-3.7.2.1I
      disk0:asr9k-k9sec-3.7.2.1I disk0:asr9k-mcast-3.7.2.1I disk0:asr9k-mgbl-3.7.2.1I
      disk0:asr9k-mpls-3.7.2.1I' started by user 'labuser' at
      18:06:32 UTC Sat Jun 03 2009.
      Info:     The changes made to software configurations will not be
      Info:     persistent across system reloads. Use the command 'admin install
      Info:     commit' to make changes persistent.
      Info:     Please verify that the system is consistent following the
      Info:     software change using the following commands:
      Info:         show system verify
```

```
--More--
```

The following example displays information for a specific install request. Use the **detail** keyword to display additional information, including impact to processes and nodes impacted.

```
RP/0/RSP0/CPU0:router(admin)# show install log 2 detail

  Install operation 2 started by user 'labuser' at 18:06:32 UTC Sat Jun 03 2009.
  install activate disk0:asr9k-diags-3.7.2.1I disk0:asr9k-k9sec-3.7.2.1I
  disk0:asr9k-mcast-3.7.2.1I disk0:asr9k-mgbl-3.7.2.1I disk0:asr9k-mpls-3.7.2.1I
  Install operation 2 completed successfully at 18:07:48 UTC Sat Jun 03 2006.

  Summary:
      Install method: parallel
      Summary of changes on nodes 0/1/SP, 0/6/SP, 0/SM0/SP, 0/SM1/SP,
      0/SM2/SP, 0/SM3/SP:
          Activated:    asr9k-diags-3.7.2.1I
          No processes affected

      Summary of changes on nodes 0/1/CPU0, 0/6/CPU0:
          Activated:    asr9k-diags-3.7.2.1I
                        asr9k-mcast-3.7.2.1I
                        asr9k-mpls-3.7.2.1I
           1 asr9k-mpls processes affected (0 updated, 1 added, 0 removed, 0 impacted)
           2 asr9k-mcast processes affected (0 updated, 2 added, 0 removed, 0 impacted)

      Summary of changes on nodes 0/RP0/CPU0, 0/RP1/CPU0:
          Activated:    asr9k-diags-3.7.2.1I
                        asr9k-k9sec-3.7.2.1I
                        asr9k-mcast-3.7.2.1I
                        asr9k-mgbl-3.7.2.1I
                        asr9k-mpls-3.7.2.1I
           6 asr9k-mgbl processes affected (0 updated, 6 added, 0 removed, 0 impacted)
           8 asr9k-mpls processes affected (0 updated, 8 added, 0 removed, 0 impacted)
           7 asr9k-k9sec processes affected (0 updated, 7 added, 0 removed, 0 impacted)
          14 asr9k-mcast processes affected (0 updated, 14 added, 0 removed, 0 impacted)

  Install logs:
      Install operation 2 'install activate disk0:asr9k-diags-3.7.2.1I
      disk0:asr9k-k9sec-3.7.2.1I disk0:asr9k-mcast-3.7.2.1I disk0:asr9k-mgbl-3.7.2.1I
      disk0:asr9k-mpls-3.7.2.1I' started by user 'labuser' at 18:06:32 UTC
      Sat Jun 03 2006.
      Info:     The changes made to software configurations will not be
      Info:     persistent across system reloads. Use the command 'admin install
      Info:     commit' to make changes persistent.
      Info:     Please verify that the system is consistent following the
      Info:     software change using the following commands:
      Info:         show system verify
      Info:         install verify packages
      Install operation 2 completed successfully at 18:07:48 UTC Sat Jun 03 2006.
```

Package Management Procedures

> **Note** Review the concepts about package management before performing the tasks described in this module.

Related Topics

Information About Package Management, on page 172

Activation and Deactivation Prerequisites

These prerequisites must be met for a package to be activated or deactivated:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Verify that all cards are installed and operating properly. For example, do not activate or deactivate packages while cards are booting, while cards are being upgraded or replaced, or when you anticipate an automatic switchover activity.
- If a ROM Monitor upgrade is required for the software package, the upgrade must be completed before the package is activated. For ROM Monitor upgrade information and procedures, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.
- Check the sanity of the configuration file system and recover from any internal inconsistencies by using the **cfs check** command.

```
RP/0/RSP0/CPU0:router# cfs check

Tue Sep 20 07:22:03.374 DST

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

- Clear any inconsistency alarms and remove any failed configurations using the **clear configuration inconsistency** command.

  An inconsistency alarm is set when there is a failure to restore the configuration; this can occur during router startup, or when a line card or route switch processor (RSP) card is inserted or removed. If an inconsistency alarm is set, a message similar to the one in this example is displayed:

```
RP/0/0/CPU0:May 26 11:58:40.662 : cfgmgr-rp[130]: %MGBL-CONFIGCLI-3
   BATCH_CONFIG_FAIL : 28 config(s) failed during startup. To view
   failed config(s) use the command - "show configuration failed startup"
```

  When the inconsistency alarm is set, all configuration commit operations fail until the alarm is cleared.

- Although more than one version of a software package can be added to a storage device, only one version of a package can be active for any card.
- Some packages require the activation or deactivation of other packages.
- The package being activated must be compatible with the current active software set.

Activation is performed only after the package compatibility checks and API version compatibility checks have been passed. If a conflict is found, an on-screen error message is displayed.

While a software package is being activated, other requests are not allowed to run on any of the impacted nodes. Package activation is completed when a message similar to this one appears:

```
Install operation 2 completed successfully at 20:30:29 UTC Mon Nov 14 2005.
```

Each CLI install request is assigned a request ID, which can be used later to review the events.

Obtaining and Placing Cisco IOS XR Software

This section contains information to locate the available software packages and to transfer them either to a local storage device or to a network server. When this is done, the package or packages can be added and activated on the router .

There are two primary ways to obtain packages in Cisco IOS XR software:

- Request the software from Cisco on a flash disk that you can insert into the removable flash disk slot (usually flash disk1:). Flash disk1: is optional. When it is installed, flash disk1: can be used to store PIE files, which can then be used to add new software to the boot device (usually flash disk0:).
- Download the Cisco IOS XR software packages to a local storage device of the DSC, such as flash disk1:, or to a remote server, such as a tftp or rcp server.

The boot device is the local disk on the DSC where Cisco IOS XR software is added and activated. PIE files should not be stored on this boot device. The default boot device is disk0:. All PIE files should be stored on flash disk1:.

Transferring Installation Files from a Network File Server to a Local Storage Device

If the Cisco IOS XR software PIE files are located on a remote TFTP, FTP, SFTP, or rcp server, you can copy the files to a local storage device such as disk1:. When the PIE files are located on a local storage device, the software packages can be added and activated on the router from that storage device. Table 29: Download Protocols Supported by Cisco IOS XR Software, on page 185 describes the supported server protocols, and the CLI syntax used copy files from each server type to the local storage device.

Tip Cisco IOS XR software PIE files can also be added to the router boot device directly from the remote server.

Note Consult your system administrator for the location and availability of your network server.

Table 29: Download Protocols Supported by Cisco IOS XR Software

Name	Description
Trivial File Transfer Protocol	TFTP allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password). It is a simplified version of FTP. **Note** Some Cisco IOS XR software images may be larger than 32 MB, and the TFTP services provided by some vendors may not support a file this large. If you do not have access to a TFTP server that supports files larger than 32 MB, download the software image using FTP or rcp.
File Transfer Protocol	FTP is part of the TCP/IP protocol stack and requires a username and password.
Remote Copy Protocol	The rcp protocol uses TCP to ensure the reliable delivery of data, and rcp downloads require a usernames.

Name	Description
SSH File Transfer Protocol	SFTP is part of the SSHv2 feature in the Security package and provides for secure file transfers. For more information, see the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*.

The router commands listed in Table 30: Commands for Copying Package Files to the Router, on page 186 show how to copy package files to the router using three types of file transfer protocols.

Table 30: Commands for Copying Package Files to the Router

Server Type	Command and Examples
TFTP	The following command syntax is used: **copy** **tftp://** *hostname_or_ipaddress* / *directory-path* / *pie-name* **disk1:** Example: `RP/0/RSP0/CPU0:router# copy tftp://10.1.1.1/images/comp-asr9k-mini.pie disk1:`
FTP	The following command syntax is used: **copy** **ftp://** *username* **:** *password* **@** *hostname_or_ipaddress* / *directory-path* / *pie-name* **disk1:** Example: `RP/0/RSP0/CPU0:router# copy ftp://john:secret@10.1.1.1/images/comp-asr9k-mini.pie disk1:`
rcp	The following command syntax is used: **copy** **rcp://** *username* **@** *hostname_or_ipaddress* / *directory-path* / *pie-name* **disk1:** Example: `RP/0/RSP0/CPU0:router# copy rcp://john@10.1.1.1/images/comp-asr9k-mini.pie disk1:`

Table 31: Command Variables for Copying and Adding Packages from a Network Server, on page 186 describes the command variables for copying packages from a network server.

Table 31: Command Variables for Copying and Adding Packages from a Network Server

Variable	Description
hostname_or_ipaddress	Host name or IP address of the server that stores the source file.
pie-name	Name of the PIE file (package). See the Overview of Cisco IOS XR Software Packages, on page 168 for descriptions of the available packages.
username	Required for FTP and rcp only and must be a valid username on the FTP or rcp server.

Variable	Description
password	Required for FTP only. If a password is not provided, the networking device accepts anonymous FTP.
directory-path	The specified directory should be a directory under the home directory of the user. In the rcp and FTP examples in Table 30: Commands for Copying Package Files to the Router, on page 186, the file being downloaded is in a subdirectory called “images” in the home directory of the user “john.” **Note** For FTP and rcp services, *directory-path* is the directory relative to the *username* home directory. If you want to specify an absolute path for the directory, you must add a "/" following the server address.

When the installation files have been transferred to a network file server or the router, you are ready to activate or upgrade the software.

Note Files with the `vm` extension are bootable installation files used only to replace all current Cisco IOS XR software. These files are installed from ROM monitor mode and cause significant router downtime. We recommend installing or upgrading software packages using PIE files only, as described in this chapter. See *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers* for information on installing from `vm` files.

Related Topics

Adding and Activating Packages, on page 198
Overview of Cisco IOS XR Software Packages, on page 168

Preparing for Software Installation Operations

This section includes instructions to prepare for software installation operations.

Note Activation is performed only after the automatic package compatibility and API version compatibility checks have been passed. If a conflict is found, an on-screen error message is displayed.

Before you begin

Before adding or activating Cisco IOS XR software:

- Update the ROM Monitor software, if necessary.
- Determine if a software change is required.
- Verify that the new package is supported on your system. Some software packages require that other packages or package versions be activated, and some packages only support specific cards.
- Review the release notes for important information related to that release and to help determine the package compatibility with your router configuration.
- Verify that the system is stable and prepared for the software changes.

SUMMARY STEPS

1. **admin**
2. **show diag**
3. Update the ROMMON software if necessary.
4. **show install active**
5. **show install pie-info** *device:package* [**brief** | **detail** | **verbose**]
6. **verify packages**
7. **exit**
8. (Optional) **show system verify start**
9. (Optional) **show system verify** [**detail** | **report**]
10. **show clock**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**show diag** **Example:** `RP/0/RSP0/CPU0:router(admin)# show diag`	Displays the ROMMON software version for all cards in the system. Verify that the correct ROMMON software version is installed before upgrading a Cisco IOS XR software package. **Note** See *Related Topics* for information regarding the required ROM Monitor (ROMMON) software version.
Step 3	Update the ROMMON software if necessary.	Updates the ROMMON software. For instructions, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.
Step 4	**show install active** **Example:** `RP/0/RSP0/CPU0:router(admin)# show install active`	Displays the active software on the router for the owner SDR. Use this command to determine what software should be added, upgraded or downgraded on the router, and to compare to the active software report after installation operations are complete. **Note** You can also display the active packages for a specific node, and view results in detailed or summary mode. See the *Software Package Management Commands on the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers* for more information.
Step 5	**show install pie-info** *device:package* [**brief** | **detail** | **verbose**]	Displays information imbedded in the package. The following keywords provide three levels of information:

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router(admin)# show install pie-info disk1:/asr9k-mcast-p.pie-3.8.30`	• **brief** (default)—Displays the expiration date of the file, the size, and the installed package name. The expiration date is used for certifying the package. • **detail**—Displays the package components, the compatible cards, the expiration date, file size, and the installed package name. • **verbose**—Displays information from the **detail** display and sub-component information. **Note** Always review the release notes for the software package for important information related to that release and to help determine the package compatibility with your router configuration.
Step 6	**verify packages** **Example:** `RP/0/RSP0/CPU0:router(admin)# install verify packages`	Verifies that there are no corrupted software files. The consistency of a previously installed software set is verified against the package file from which it originated. This command can be used as a debugging tool to verify the validity of the files that constitute the packages, to determine if there are any corrupted files. This command also checks for corruptions of installation state files and MBI image files. This command is particularly useful when issued after the activation of a package or upgrading the Cisco IOS XR software to a major release. **Note** The **install verify packages** command can take up to two minutes per package to process.
Step 7	**exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits administration EXEC mode and returns to EXEC mode.
Step 8	(Optional) **show system verify start** **Example:** `RP/0/RSP0/CPU0:router# show system verify start`	Starts the system status check.
Step 9	(Optional) **show system verify** [**detail** | **report**] **Example:** `RP/0/RSP0/CPU0:router# show system verify`	Displays system status information. A variety of information is displayed including the memory and CPU usage, process status, protocol status, and other status information. Use this information to verify that the system is stable. • **detail**—Displays additional information at the card and processor level, including actual numbers. • **report**—Displays the same information as the default **show system verify** command

	Command or Action	Purpose
		Note Although most of the output should display the status "OK," some processes may show other output, such as "Warning." This does not specifically indicate a problem. Contact your Cisco technical support representative for more information on the output of this command.
Step 10	**show clock** **Example:** `RP/0/RSP0/CPU0:router# show clock`	Verifies that the system clock is correct. Software operations use certificates based on router clock times.

Related Topics

Activation and Deactivation Prerequisites, on page 184

Examples

Verifying That the ROM Monitor Version Is Correct: Example

In the following example, the ROM Monitor software version is displayed in the "ROMMON:" field for each card.

Note For instructions to upgrade the ROM Monitor software, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# show diag

Mon Jun 22 12:55:10.554 PST

NODE module 0/RSP0/CPU0 :

  MAIN:  board type 0x100302
  S/N:   FOC1230803H
  Top Assy. Number:   68-3160-04
  PID:   A2K-RSP-4G-HDD=
  UDI_VID:   VP4
  HwRev: V4.8
  New Deviation Number: 0
  CLEI:  IPUCARJBAA
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  MONLIB: QNXFFS Monlib Version 3.2
  ROMMON: Version 1.0(20081208:173612) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
    Compact Flash : V1.0
     XbarSwitch0  : V1.3
     XbarSwitch1  : V1.3
     XbarArbiter  : V1.0
     XbarInterface  : V0.0
```

```
      IntCtrl  : V1.14
      ClkCtrl  : V1.13
      PuntFPGA  : V1.5
      HD  : V3.0
      USB0  : V77.20
      USB1  : V77.20
      CPUCtrl  : V1.17
      UTI  : V1.6
      LIU  : V1.0
      MLANSwitch  : V0.0
      EOBCSwitch  : V2.0
      CBC (active partition)  : v1.2
      CBC (inactive partition)  : v1.1

NODE fantray 0/FT0/SP :

  MAIN:  board type 0x900211
  S/N:
  Top Assy. Number:   32-0000-00
  PID:
  UDI_VID:
  HwRev: V32.0
  New Deviation Number: 0
  CLEI:
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:
     CBC (active partition)  : v4.0
     CBC (inactive partition)  : v0.13

NODE fantray 0/FT1/SP :

  MAIN:  board type 0x900211
  S/N:
  Top Assy. Number:   32-0000-00
  PID:
  UDI_VID:
  HwRev: V32.0
  New Deviation Number: 0
  CLEI:
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:
     CBC (active partition)  : v4.0
     CBC (inactive partition)  : v0.13

NODE module 0/1/CPU0 :

   MAIN:  board type 0x20207
  S/N:   FOC123081J6
  Top Assy. Number:   68-3182-03
  PID:   A9K-40GE-B
  UDI_VID:   V1D
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:
  Board State : IOS XR RUN
  PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
  ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
  Board FPGA/CPLD/ASIC Hardware Revision:
     NP0  : V3.194
     NP1  : V3.194
     NP2  : V3.194
     NP3  : V3.194
```

```
      XbarInterface  : V18.4
      Bridge0  : V0.38
      Bridge1  : V0.38
      CPUCtrl  : V0.15
      USB  : V77.20
      PortCtrl  : V0.8
      PHYCtrl  : V0.6
      40 Port Gigabit Ethernet Daughter board  : V0.0
      CBC (active partition)  : v2.2
      CBC (inactive partition)  : v2.1

NODE module 0/4/CPU0 :

   MAIN:  board type 0x2020a
   S/N:   FOC123081JA
   Top Assy. Number:   68-3183-02
   PID:   A9K-8T/4-B
   UDI_VID:   V1D
   HwRev: V0.0
   New Deviation Number: 0
   CLEI:  IPU3AE0CAA
   Board State : IOS XR RUN
   PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
   ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
   Board FPGA/CPLD/ASIC Hardware Revision:
      NP0  : V3.194
      NP1  : V3.194
      NP2  : V3.194
      NP3  : V3.194
      XbarInterface  : V18.4
      Bridge0  : V0.38
      Bridge1  : V0.38
      CPUCtrl  : V0.15
      USB  : V77.20
      PortCtrl  : V0.10
      PHYCtrl  : V0.7
      PHY0  : V0.16
      PHY1  : V0.16
      PHY2  : V0.16
      PHY3  : V0.16
      PHY4  : V0.16
      PHY5  : V0.16
      PHY6  : V0.16
      PHY7  : V0.16
      8 Port Ten Gigabit Ethernet Daughter board  : V0.0
      CBC (active partition)  : v2.2
      CBC (inactive partition)  : v2.1

NODE module 0/6/CPU0 :

   MAIN:  board type 0x20208
   S/N:   FHH12250033
   Top Assy. Number:   68-3184-02
   PID:   A9K-4T-B
   UDI_VID:   V1D
   HwRev: V0.0
   New Deviation Number: 0
   CLEI:
   Board State : IOS XR RUN
   PLD:    Motherboard: N/A, Processor: 0x8004 (rev: 2.2), Power: N/A
   ROMMON: Version 1.0(20081208:174521) [ASR9K ROMMON]
   Board FPGA/CPLD/ASIC Hardware Revision:
      NP0  : V3.194
      NP1  : V3.194
```

```
     NP2  : V3.194
     NP3  : V3.194
     XbarInterface  : V18.4
     Bridge0  : V0.38
     Bridge1  : V0.38
     CPUCtrl  : V0.15
     USB  : V77.20
     PHY0  : V0.16
     PHY1  : V0.16
     PHY2  : V0.16
     PHY3  : V0.16
     PortCtrl  : V0.10
      PHYCtrl  : V0.7
     4 Port Ten Gigabit Ethernet Daughter board  : V0.0
     CBC (active partition)  : v2.2
     CBC (inactive partition)  : v2.1

NODE power-module 0/PM0/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:

NODE power-module 0/PM1/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:

NODE power-module 0/PM2/SP :

  MAIN:  board type 0xf00188
  S/N:
  Top Assy. Number:   341-00032-01
  PID:   A9K-3KW-AC
  UDI_VID:   V00
  HwRev: V0.0
  New Deviation Number: 0
  CLEI:  ACACACACAC
  PLD:    Motherboard: N/A, Processor: N/A, Power: N/A
  ROMMON:
  Board FPGA/CPLD/ASIC Hardware Revision:

Rack 0 - ASR-9010 Chassis, Includes Accessories
  RACK NUM: 0
  S/N:
  PID:   ASR-9010 Backplane
  VID:   0.1
```

```
  Desc:  ASR-9010 Chassis, Includes Accessories
  CLEI:  NOCLEI
  Top Assy. Number:   68-1234-56
```

Displaying the Active Software for the Entire System: Example

The following example displays the active packages for the entire system. Use this information to determine if a software change is required:

```
RP/0/RSP0/CPU0:router(admin)# show install active summary

Mon Jun 22 13:01:46.438 PST
Default Profile:
  SDRs:
    Owner
  Active Packages:
    disk0:comp-asr9k-mini-3.9.0.12I
    disk0:asr9k-fpd-3.9.0.12I
    disk0:asr9k-k9sec-3.9.0.12I
    disk0:asr9k-mcast-3.9.0.12I
    disk0:asr9k-mgbl-3.9.0.12I
    disk0:asr9k-mpls-3.9.0.12I
```

Displaying Information About the Contents of a PIE File: Example

In the following example, information is displayed about the manageability PIE. This command displays the expiry date of the package, the cards supported by the package, and other details. Use this information to verify the compatibility of the package with your system and other software packages.

Note A software activation is performed only after the automatic package compatibility and API version compatibility checks have been passed. If a conflict is found, an on-screen error message is displayed.

```
RP/0/RSP0/CPU0:router(admin)# show install pie-info disk1:/
asr9k-mgbl-p.pie-3.8.0 detail

Contents of pie file '/disk1:/asr9k-mgbl-p.pie-3.8.0':
    Expiry date       : Jan 19, 2007 02:55:56 UTC
    Uncompressed size : 17892613

    asr9k-mgbl-3.8.0
        asr9k-mgbl V3.8.0[00]  Manageability Package
        Vendor : Cisco Systems
        Desc   : Manageability Package
        Build  : Built on Wed May 10 08:04:58 UTC 2006
        Source : By edde-bld1 in /vws/aga/production/3.8.0/asr9k/workspace for c28
        Card(s): RP, DRP, DRPSC
        Restart information:
          Default:
            parallel impacted processes restart
```

```
        Components in package asr9k-mgbl-3.8.0, package  asr9k-mgbl:
            manageability-cwi V[r33x/2]  Craft Web Interface related binaries ae
            asr9k-feature-ipsla V[r33x/1]  IPSLA time stamping feature
            doc-asr9k-mgbl V[r33x/2]  Contains the man page documentation for asr9ks
 --More--
```

Verifying That There Are No Corrupted Software Files: Example

The following sample output verifies the consistency of the currently active software against the file from which it originated:

```
RP/0/RSP0/CPU0:router(admin)# install verify packages

Mon Jun 22 13:19:08.590 PST
Install operation 3 '(admin) install verify packages' started by user 'user'
via CLI at 13:19:08 DST Mon Jun 22 2009.
The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#Info:
This operation can take up to 2 minutes per package being verified.
Info:     Please be patient.

Info:     0/6/CPU0 [LC] [SDR: Owner]
Info:         meta-data: [SUCCESS] Verification Successful.
Info:         /install/asr9k-scfclient-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-os-mbi-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-mpls-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mcast-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-lc-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fwdg-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fpd-3.9.0.12I: [ERROR] Detected anomalies.
Info:         /install/asr9k-diags-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-base-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-admin-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:     0/1/CPU0 [LC] [SDR: Owner]
Info:         meta-data: [SUCCESS] Verification Successful.
Info:         /install/asr9k-scfclient-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-os-mbi-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-mpls-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mcast-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-lc-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fwdg-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fpd-3.9.0.12I: [ERROR] Detected anomalies.
Info:         /install/asr9k-diags-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-base-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-admin-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:     0/4/CPU0 [LC] [SDR: Owner]
Info:         meta-data: [SUCCESS] Verification Successful.
Info:         /install/asr9k-scfclient-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-os-mbi-3.9.0.12I: [SUCCESS] Verification
```

```
Info:     Successful.
Info:         /install/asr9k-mpls-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mcast-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-lc-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fwdg-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fpd-3.9.0.12I: [ERROR] Detected anomalies.
Info:         /install/asr9k-diags-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-base-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-admin-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:     0/RSP0/CPU0 [RP] [SDR: Owner]
Info:         meta-data: [SUCCESS] Verification Successful.
Info:         /install/asr9k-fpd-3.9.0.12I: [ERROR] Detected anomalies.
Info:         /install/asr9k-mpls-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mgbl-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-mcast-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-k9sec-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-os-mbi-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-base-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-admin-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-fwdg-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-lc-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-rout-3.9.0.12I: [SUCCESS] Verification Successful.
Info:         /install/asr9k-diags-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:         /install/asr9k-scfclient-3.9.0.12I: [SUCCESS] Verification
Info:     Successful.
Info:     Verification Summary:
Info:         0/6/CPU0: ERROR. Anomalies found.
Info:         0/1/CPU0: ERROR. Anomalies found.
Info:         0/4/CPU0: ERROR. Anomalies found.
Info:         0/RSP0/CPU0: ERROR. Anomalies found.
Info:         Anomalies found on the primary RP.
Info:         No standby RP is present.
Info:         Please contact your technical services representative to repair
Info:     the system.
Install operation 3 completed successfully at 13:21:29 DST Mon Jun 22 2009.
```

Verifying the Current System Status: Example

The following example shows how to prepare for system verification:

```
RP/0/RSP0/CPU0:router# show system verify start

  Storing initial router status ...
  done.
```

The following example shows output from running the **show system verify** command.

Note Although most of the output should display the status "OK," some processes may show other output, such as "Warning." This does not specifically indicate a problem. Contact your Cisco technical support representative for more information on the output of this command.

```
RP/0/RSP0/CPU0:router# show system verify

  Getting current router status ...
  System Verification Report
  ==========================
  - Verifying Memory Usage
  - Verified Memory Usage                                    : [OK]
  - Verifying CPU Usage
  - Verified CPU Usage                                       : [OK]

  - Verifying Blocked Processes
  - Verified Blocked Processes                               : [OK]
  - Verifying Aborted Processes
  - Verified Aborted Processes                               : [OK]
  - Verifying Crashed Processes
  - Verified Crashed Processes                               : [OK]

  - Verifying LC Status
  - Verified LC Status                                       : [OK]
  - Verifying QNET Status
  Unable to get current LC status info
  - Verified QNET Status                                     : [FAIL]

  - Verifying GSP Fabric Status
  - Verified GSP Fabric Status                               : [OK]
  - Verifying GSP Ethernet Status
    gsp WARNING messages for router
    Current set of gsp ping nodes does not match initial set of nodes
  - Verified GSP Ethernet Status                             : [WARNING]

  - Verifying POS interface Status
  - Verified POS interface Status                            : [OK]
  - Verifying TenGigE interface Status
  - Verified TenGigE interface Status                        : [OK]

  - Verifying TCP statistics
  - Verified TCP statistics                                  : [OK]
  - Verifying UDP statistics
    tcp_udp_raw WARNING messages for router
    UDP Packets sent has not increased during this period.
  - Verified UDP statistics                                  : [WARNING]
  - Verifying RAW statistics
  - Verified RAW statistics                                  : [OK]

  - Verifying RIB Status
  - Verified RIB Status                                      : [OK]
  - Verifying CEF Status
  - Verified CEF Status                                      : [OK]
  - Verifying CEF Consistency Status
  - Verified CEF Consistency Status                          : [OK]
  - Verifying BGP Status
  - Verified BGP Status                                      : [OK]
  - Verifying ISIS Status
  - Verified ISIS Status                                     : [OK]
  - Verifying OSPF Status
```

```
- Verified OSPF Status                                      : [OK]

- Verifying Syslog Messages
- Verified Syslog Messages                                  : [OK]

System may not be stable. Please look into WARNING messages.
```

Verifying That the System Clock Is Correct: Example

The following example displays the current system clock setting:

```
RP/0/RSP0/CPU0:router# show clock

02:14:51.474 PST Wed Jan 28 2009
```

Adding and Activating Packages

The procedure in this section describes how to upgrade or add Cisco IOS XR software PIE files that are stored on a local storage device, such as a flash disk, or on a remote TFTP, FTP, SFTP, or rcp server. The PIE software file can include any of the following:

- The Cisco IOS XR Unicast Routing Core Bundle (six packages in one composite PIE file)
- Any of the optional packages (one package per PIE file)
- Software maintenance upgrades (SMUs)

When you need to add and activate two or more of the preceding package types, you should add and activate them in the order listed above.

Note When adding and activating two or more packages, optional packages can be activated together. Also, if the operation is a reload, multiple packages can be activated together. For example, five reload SMUs can be activated together or the Cisco IOS XR Unicast Routing Core Bundle plus the SMUs and optional packages can be activated together.

For a description of the software management process, see the *Related Topics* section.

These instructions are also used to downgrade software packages.

Note By default, installation operations are performed asynchronously: the CLI prompt is returned before the operation is complete, allowing the operator to continue work while the installation is completed in the background. Use the **synchronous** keyword at the end of install commands to delay the return of the CLI prompt until an installation operation is complete. See the *Related Topics* section for more information.

Before you begin

Before upgrading or adding packages, verify that these prerequisites have been met:

- Verify that the ROMMON version is correct. For instructions on upgrading ROM Monitor, see *ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*.
- All packages to be upgraded or added are present on a local storage device (for example a flash disk), or a network file server.
- Prerequisites for the activation of packages are met as described in the Prerequisites section.
- Complete the procedures described in the Preparing for Software Installation Operations, on page 187 section.

Note To use the automatic FPD upgrade feature, the **fpd auto-upgrade** command must be enabled in administration configuration mode.

SUMMARY STEPS

1. Connect to the console port and log in.
2. (Optional) **dir** *flash-disk* **:**
3. **admin**
4. **install add** [**source** *source-path* | **tar**] file [**activate**]
5. (Optional) **show install inactive summary**
6. **install activate** {**id** *add-id* | *device package*} [**test**] [**location** *node-id*] [**pause sw-change**] [**prompt-level** {**all** | **none**}] [**auto-abort-timer** {*time* | **off**}]
7. Repeat Step 4, on page 200 through Step 6, on page 201 until all packages are activated.
8. (Optional) **show install active summary**
9. (Optional) **install verify packages**
10. (Optional) **exit**
11. (Optional) **show system verify start**
12. **admin**
13. (Optional) **install commit**
14. Upgrade the field-programmable device (FPD) software, if necessary.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Connect to the console port and log in.	Establishes a CLI management session with the SDR. Connect to the console port for the active DSC. For more information on console connections, see *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*.
Step 2	(Optional) **dir** *flash-disk* **:** **Example:** `RP/0/RSP0/CPU0:router# dir disk1:`	Displays the package files that are available for package upgrades and additions. **Note** Only PIE files can be added and activated using this procedure.
Step 3	Required: **admin** **Example:**	Enters administration EXEC mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# admin`	**Note** Some **show install** commands can be entered in EXEC mode on an SDR.

Step 4	**install add** [**source** *source-path* | **tar**] file [**activate**]
	Example:
	`RP/0/RSP0/CPU0:router(admin)# install add disk1:/asr9k-mgbl-px.pie-6.0.1`
	or
	`RP/0/RSP0/CPU0:router(admin)# install add source`
	`tftp://10.1.1.1/images/ asr9k-k9sec-p.pie asr9k-mpls-p.pie asr9k-mcast-p.pie`
	or
	`RP/0/RSP0/CPU0:router(admin)# install add`
	`ftp://john:secret@10.1.1.1/images/asr9k-k9sec-p.pie`
	or
	`RP/0/RSP0/CPU0:router(admin)# install add tar rcp://john@10.1.1.1/images/asr9k-iosxr-3.6.0.tar`
		Unpacks a PIE file from local storage device or network server and adds the package files to the boot device of the router. The boot device is located on the DSC.
		• If the **source** keyword is used, the *source-path* specifies the directory path that is used for multiple filenames in the same directory.
		• If the **tar** keyword is used, all PIE files contained in the tar file are unpacked.
		The *file* argument can take any of these formats:
		• *device filename*
		• **tftp://***hostname_or_ipaddress /directory-path /filename*
		• **ftp://** *username***:***password***@***hostname_or_ipaddress /directory-path/filename*
		• **rcp://***username***@***hostname_or_ipaddress/ directory-path/filename*
		These are descriptions for each of the terms used here:
		• *device*—Name of the local storage device where the PIE file is stored, such as **disk1:**/.
		• *filename*—Name of the PIE file you want to add. If the **tar** keyword is used, the *file* argument is the name of a tar file containing one or more PIE files, or directories containing PIE files.
		• **tftp://**—Unpacks the PIE file from a network server using Trivial File Transfer Protocol.
		• **ftp://**—Unpacks the PIE file from a network server using File Transfer Protocol.
		• **rcp://**—Unpacks the PIE file from a network server using Remote Copy Protocol
		• *hostname_or_ipaddress*—Host name or IP address of the network file server.
		• *directory-path*—Network file server path that leads to the PIE file to be added.
		• *username*—Username of user that has access privileges to the directory in which the PIE file is stored.
		• *password*—Password associated with the username of user that has access privileges to the directory in which the PIE file is stored.

	Command or Action	Purpose
		• **activate**—Automatically activates the software package after it is successfully added. **Note** Multiple versions of a software package can be added to the storage device without impacting the running configuration, but only one version of a package can be activated for a card. **Tip** The automatic FPD upgrade occurs only when the FPD pie is added and activated together with the install PIE.
Step 5	(Optional) **show install inactive summary** **Example:** `RP/0/RSP0/CPU0:router(admin)# show install inactive summary`	Displays the inactive packages on the router. Verify that the package added in the previous step appears in the display.
Step 6	**install activate** {**id** *add-id* | *device package*} [**test**] [**location** *node-id*] [**pause sw-change**] [**prompt-level** {**all** | **none**}] [**auto-abort-timer** {*time* | **off**}] **Example:** `RP/0/RSP0/CPU0:router(admin)# install activate disk0:asr9k-mini-px-4.3.99`	Activates a package that was added to the router. (Skip this step if the package was activated earlier with the **install add** command.) • **id** *add-id*—Specifies the package using the operation ID of the **install add** operation in which you added the package. The operation ID is provided in the output of the **install add** command. You can also use **show install log** to display installation operation IDs. • *device:package*—Specifies the package by name. Replace the *device:package* argument with the name of the boot device and inactive package, which can be displayed as described in the previous step. **Note** Press **?** after a partial package name to display all possible matches available for activation. If there is only one match, press [TAB] to fill in the rest of the package name. • **location** *node-id*—Activates a package for a specific card (node). To display a list of node IDs for the entire system, enter the **show platform** command in administration EXEC mode. A package cannot be activated on a single node unless some version of the package being activated is already active on all nodes. **Note** By default, packages are activated for all cards supported by that package. • **pause sw-change**—Pauses the operation after preparatory checks and before the configuration is locked for the actual activation. This action enables you to hold the operation while you perform

	Command or Action	Purpose
		configuration changes, and proceed with the activation whenever you choose. This operation is useful, for example, if your workflow involves configuring a router out of the network during software installation and you want to minimize the time that the router is out of the network. Follow onscreen instructions to control the pausing and completion of the operation. • **prompt-level**—Use a prompt-level of **all** to view all stages of the installation process and to specify whether to continue, or not. • **auto-abort-timer**—Specifies an abort timer value, in minutes, which when expired loads the last committed loadpath. The default is 60. The timer is enabled by default. After the installation, if the activated software is working correctly, use the **install commit** command to cancel the timer and commit the new loadpath. **Note** The package being activated must be compatible with the currently active software to operate. When an activation is attempted, the system runs an automatic compatibility check to ensure that the package is compatible with the other active software on the router. The activation is permitted only after all compatibility checks have been passed. **Tip** When activating packages, use the **test** option to test the effects of a command without impacting the running system. After the activation process finishes, enter the **show install log** command to display the process results. **Tip** The automatic FPD upgrade occurs only when the FPD pie is added and activated together with the install PIE.
Step 7	Repeat Step 4, on page 200 through Step 6, on page 201 until all packages are activated.	Activates additional packages as required.
Step 8	(Optional) **show install active summary** **Example:** `RP/0/RSP0/CPU0:router(admin)# show install active`	Displays all active packages. Use this display to determine if the correct packages are active:
Step 9	(Optional) **install verify packages** **Example:**	Verifies the consistency of a installed software set with the package file from which it originated. This command can be used as a debugging tool to verify the validity of

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(admin)# install verify packages`	the files that constitute the packages, to determine whether there are any corrupted files. This command also checks for corruptions of installation state files and MBI image files. This command is particularly useful when issued after the activation of a package or upgrading the Cisco IOS XR software to a major release. **Note** The **install verify packages** command can take up to two minutes for each package to process.
Step 10	(Optional) **exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits administration EXEC mode and returns to EXEC mode.
Step 11	(Optional) **show system verify start** **Example:** `RP/0/RSP0/CPU0:router# show system verify start`	Starts the system status check.
Step 12	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

Step 13	(Optional) **install commit**
	Example:
	`RP/0/RSP0/CPU0:router# dir disk1:`
	`RP/0/RSP0/CPU0:router(admin)# install commit`
		Commits the current set of packages on the router so that these packages are used if the router is restarted.
		For more information, see the *Related Topics* section.

Step 14	Upgrade the field-programmable device (FPD) software, if necessary.	Whenever a Cisco IOS XR software image that supports SPAs and SIPs is released, a companion SPA or SIP FPD image is bundled with the Cisco IOS XR software release. Generally, the FPD image is not automatically upgraded. You must manually upgrade the FPD image running on the SPA or SIP when you upgrade the Cisco IOS XR software image. FPD versions must be compatible with the Cisco IOS XR software that is running on the router. **Note** If you have enabled the **fpd auto-upgrade** command and add and activate the FPD PIE together with the software installation PIE, the FPD image is automatically upgraded before the router is rebooted.

	Command or Action	Purpose
		For information on FPDs, including instructions to upgrade FPD images, see the *Upgrading FPD Cisco IOS XR Software* section.

Related Topics

Obtaining and Placing Cisco IOS XR Software, on page 185
Activation and Deactivation Prerequisites, on page 184
Preparing for Software Installation Operations, on page 187
Information About Package Management, on page 172
Downgrading Packages, on page 179
PIE Filenames and Version Numbers, on page 170
Committing the Active Package Set, on page 206
Prerequisites for ISSU
Restrictions for ISSU
ISSU Software Images

Examples

Adding a Package: Example

The following example shows how to add the contents of a PIE file on disk1: to the boot device. Because the software package is added to the boot device by default, it is not necessary to specify the destination device in the CLI.

Note From Cisco IOS XR Software Release 6.0.1 and later, you must append a forward slash ('/') to the source location (for example, **disk1:/**) of the PIE file in the **install add** command.

```
RP/0/RSP0/CPU0:router(admin)# install add disk1:/asr9k-mpls-px.pie-6.0.1 synchronous

Install operation 4 'install add /disk1:/asr9k-mpls.pie-mpls-px.pie-6.0.1 synchronous'
started by user
'cisco' at 18:10:18 UTC Sat Apr 08 2009.
Info:     The following package is now available to be activated:
Info:
Info:         disk0:asr9k-mpls-px.pie-6.0.1
Info:
Install operation 4 completed successfully at 18:14:11 UTC Sat Apr 08 2009.
```

The following example shows how to add the contents of a PIE file on a TFTP server to the boot device:

```
RP/0/RSP0/CPU0:router(admin)# install add tftp://209.165.201.1/
asr9k-mpls.pie synchronous

Install operation 4 '(admin) install add /tftp://209.165.201.1/asr9k-mpls.pie synchronous'

  started by user 'cisco' at 18:16:18 UTC Thu Jan 03 2009.
```

```
Info:     The following package is now available to be activated:
Info:
Info:         disk0:asr9k-mpls-3.7.2
Info:
Install operation 4 completed successfully at 18:19:10 UTC Thu Jan 03 2009.
```

Activating a Package: Example

The following example shows the activation of the MPLS package. The package is activated on the boot device disk0:.

```
RP/0/RSP0/CPU0:router(admin)# install activate disk0:
asr9k-mpls-3.7.2 synchronous

Install operation 15 'install activate disk0:asr9k-mpls-3.7.2 synchronous'
started by user 'lab' at 19:15:33 UTC Sat Apr 08 2009.
Info:     The changes made to software configurations will not be persistent
Info:     across system reloads. Use the command 'admin install commit' to make
Info:     changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
Install operation 5 completed successfully at 19:16:18 UTC Sat Apr 08 2009.
```

Activating a Package by Specifying an Operation ID: Example

The following example shows the activation of the MPLS package using the operation ID of the **install add** operation that added the package:

```
RP/0/RSP0/CPU0:router(admin)# install activate id 4

Install operation 5 '(admin) install activate id 4' started by user 'lab' via
CLI at 18:20:17 UTC Thu Jan 03 2009.
Info:     This operation will activate the following package:
Info:         disk0:asr9k-mpls-3.7.2
Info:     Install Method: Parallel Process Restart
The install operation will continue asynchronously.
Info:     The changes made to software configurations will not be persistent
Info:     across system reloads. Use the command '(admin) install commit' to
Info:     make changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
Install operation 5 completed successfully at 18:21:30 UTC Thu Jan 03 2009.
```

Adding and Activating a Package from an FTP File Server with One Command: Example

To add and activate a package with a single command, enter the **install add** command with the **activate** keyword. In the following example, the Manageability PIE located on disk1: is verified, unpacked, and added to the boot device disk0. Because this operation is performed in administration EXEC mode, the package is activated for all SDRs in the system.

```
RP/0/RSP0/CPU0:router(admin)# install add disk1:/
asr9k-mgbl-px.pie-6.0.1 activate

Install operation 4 'install add /disk1:/asr9k-mgbl-px.pie-6.0.1 activate' started
by user 'cisco' at 07:58:56 UTC Wed Mar 01 2009.
The install operation will continue asynchronously.
:router(admin)#Part 1 of 2 (add software): Started
Info:     The following package is now available to be activated:
Info:
Info:         disk0:asr9k-mgbl-px.pie-6.0.1
Info:
Part 1 of 2 (add software): Completed successfully
Part 2 of 2 (activate software): Started
Info:     The changes made to software configurations will not be
  persistent across system reloads. Use the command 'admin install
Info:     commit' to make changes persistent.
Info:     Please verify that the system is consistent following
  the software change using the following commands:
Info:         show system verify
Info:         install verify packages
Part 2 of 2 (activate software): Completed successfully
Part 1 of 2 (add software): Completed successfully
Part 2 of 2 (activate software): Completed successfully
Install operation 4 completed successfully at 08:00:24 UTC Wed Mar 01 2009.
```

Displaying the Active Packages: Example

The following example displays a summary of the active packages on a router. Because this operation is performed in administration EXEC mode, the active packages for all SDRs are displayed.

```
RP/0/RSP0/CPU0:router(admin)# show install active summary
Mon Jun 22 23:41:19.509 PST
Default Profile:
  SDRs:
    Owner
  Active Packages:
    disk0:comp-asr9k-mini-3.9.0.12I
    disk0:asr9k-fpd-3.9.0.12I
    disk0:asr9k-k9sec-3.9.0.12I
    disk0:asr9k-mcast-3.9.0.12I
    disk0:asr9k-mgbl-3.9.0.12I
    disk0:asr9k-mpls-3.9.0.12I
```

Committing the Active Package Set

When a package is activated, it becomes part of the current running configuration. To make the package activation persistent across system-wide reloads, enter the **install commit** command. On startup, DSC of the owner SDR loads this committed software set. If the system is reloaded before the current active software is committed with the **install commit** command, the previously committed software set is used.

If the system is reloaded before the current active software is committed with the **install commit** command, the previously committed software set is used.

> **Tip** Before committing a package set, verify that the SDR is operating correctly and is forwarding packets as expected.

SUMMARY STEPS

1. **admin**
2. **install commit**
3. **show install committed** [**detail** | **summary** | **verbose**] [**location** *node-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	Required: **install commit** **Example:** `RP/0/RSP0/CPU0:router(admin)# install commit`	Commits the current set of packages on the router so that these packages are used if the router is restarted.
Step 3	**show install committed** [**detail** | **summary** | **verbose**] [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router(admin)# show install committed`	Displays which packages are committed.

Examples

Committing the Active Package Set: Example

In the following example, the active software packages are committed on the router:

```
RP/0/RSP0/CPU0:router(admin)# install commit

Install operation 16 'install commit' started by user 'lab' at 19:18:58 UTC
Sat Apr 08 2009.
Install operation 16 completed successfully at 19:19:01 UTC Sat Apr 08 2009.
```

Displaying the Committed Package Versions: Example

In the following example, the committed packages are shown for the owner SDR:

```
RP/0/RSP0/CPU0:router(admin)# show install committed

Tue Jun 23 05:11:29.968 PST
Secure Domain Router: Owner

  Node 0/RSP0/CPU0 [RP] [SDR: Owner]
    Boot Device: disk0:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/mbiasr9k-rp.vm
    Committed Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
      disk0:asr9k-fpd-3.9.0.12I
      disk0:asr9k-k9sec-3.9.0.12I
      disk0:asr9k-mcast-3.9.0.12I
      disk0:asr9k-mgbl-3.9.0.12I
      disk0:asr9k-mpls-3.9.0.12I

  Node 0/1/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Committed Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
      disk0:asr9k-fpd-3.9.0.12I
      disk0:asr9k-mcast-3.9.0.12I
      disk0:asr9k-mpls-3.9.0.12I

  Node 0/4/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Committed Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
      disk0:asr9k-fpd-3.9.0.12I
      disk0:asr9k-mcast-3.9.0.12I
      disk0:asr9k-mpls-3.9.0.12I

  Node 0/6/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Committed Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
      disk0:asr9k-fpd-3.9.0.12I
      disk0:asr9k-mcast-3.9.0.12I
      disk0:asr9k-mpls-3.9.0.12I
```

As with the **show install active** command, the **show install committed** command may display a composite package that represents all packages in the Cisco IOS XR Unicast Routing Core Bundle.

Upgrading to Cisco IOS XR Software Release 4.0

In Cisco IOS XR Software Release 4.0, the software packages were reorganized into functionally well-defined and independently-releasable packages. For this reason, when you upgrade from a software release prior to Release 4.0, you must perform the following procedure in order to synchronize all of the software packages according to the reorganized structure. General information regarding the the addition and activation of software packages is not covered in this procedure.

The main difference between the standard upgrade procedure and the procedure required to upgrade from Release 3.x to 4.x is that the later requires the addition of one additional software package, known as the *upgrade package* (asr9k-upgrade-p.pie).

Before you begin

Before performing this procedure, see the adding and activating software package procedures described in this module.

SUMMARY STEPS

1. **admin**
2. **install add tftp://** *hostname_or_ipaddress* / *directory-path* / *mandatory-bundle-pie*
3. **install add tftp://** *hostname_or_ipaddress* / *directory-path* / **asr9k-upgrade-p.pie**
4. **install activate** *device***:***mandatory-bundle-pie* *device***:***upgrade-package*
5. **install deactivate** *device***:***upgrade-package*
6. (Optional) **install commit**
7. **install remove** *device***:***upgrade-package*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Required: **admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**install add tftp://** *hostname_or_ipaddress* / *directory-path* / *mandatory-bundle-pie* **Example:** `RP/0/RSP0/CPU0:router(admin)# install add tftp://10.1.1.1/auto/tftpboot/usr/400/asr9k-mini-p.pie`	Unpacks the mandatory bundle PIE file from a network server and adds the package file to the boot device of the router. **Note** Refer to the standard procedure to add and activate packages to see other options of PIE file locations and a description of the various arguments for the **install add** command.
Step 3	**install add tftp://** *hostname_or_ipaddress* / *directory-path* / **asr9k-upgrade-p.pie** **Example:** `RP/0/RSP0/CPU0:router(admin)# install add tftp://10.1.1.1/auto/tftpboot/usr/400/asr9k-upgrade-p.pie`	Unpacks the upgrade PIE file from a network server and adds the package file to the boot device of the router.
Step 4	**install activate** *device***:***mandatory-bundle-pie* *device***:***upgrade-package* **Example:** `RP/0/RSP0/CPU0:router(admin)# install activate disk0:asr9k-mini-p-4.0.0 disk0:asr9k-upgrade-p-4.0.0`	Activates the package that was added to the router together with the upgrade package. **Note** The bundle of mandatory packages and the upgrade bundle are activated together to perform the successful upgrade from release 3.x to 4.x.

	Command or Action	Purpose
Step 5	**install deactivate** *device***:***upgrade-package* **Example:** `RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:asr9k-upgrade-p-4.0.0`	Deactivates the upgrade package on the router . For specific information regarding the deactivation and removal of software packages, refer to the general procedure.
Step 6	(Optional) **install commit** **Example:** `RP/0/RSP0/CPU0:router(admin)# install commit`	Commits the current set of packages so that these packages are used if the router is restarted. Packages can be removed only if the deactivation operation is committed.
Step 7	Required: **install remove** *device***:***upgrade-package* **Example:** `RP/0/RSP0/CPU0:router(admin)# install remove disk0:asr9k-upgrade-p-4.0.0`	Removes the inactive upgrade package.

Example

The following example illustrates the upgrade operation:

```
RP/0/RSP0/CPU0:router(admin)# install add /tftp://223.255.254.254/auto/tftpboot/users/user/
asr9k-mini-p.pie

Fri Jul  9 03:53:11.052 UTCRP/0/RP1/CPU0:Jul  9 03:53:12.053 :
instdir[235]: %INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Install operation 4 '(admin) install add
/tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-mini-p.pie'
started by user 'lab'
Install operation 4 '(admin) install add
/tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-mini-p.pie'
started by user 'lab' via CLI at 03:53:12 UTC Fri Jul 09 2010.
The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#
Info:    The following package is now available to be activated:
Info:    disk0:asr9k-mini-p-4.0.0
Info:    The package can be activated across the entire router.
Info:    RP/0/RP1/CPU0:Jul  9 04:32:26.152 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_COMPLETED_SUCCESSFULLY :
Info:    Install operation 4 completed successfully
Info:    Install operation 4 completed successfully at 04:32:26 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)# install add /tftp://223.255.254.254/auto/tftpboot/users/user/
asr9k-mpls-p.pie

Fri Jul  9 05:07:52.237 UTCRP/0/RP1/CPU0:Jul  9 05:07:53.710 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Info:    Install operation 5 '(admin) install add
Info:    /tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-mpls-p.pie'
Info:    started by user 'lab'
Info:    Install operation 5 '(admin) install add
Info:    /tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-mpls-p.pie'
Info:    started by user 'lab' via CLI at 05:07:53 UTC Fri Jul 09 2010.
Info:    The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#
```

```
Info:    RP/0/RP1/CPU0:Jul  9 05:09:08.854 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_COMPLETED_SUCCESSFULLY :
Install operation 5 completed successfully
Info:    The following package is now available to be activated:
Info:    disk0:asr9k-mpls-p-4.0.0
Info:    The package can be activated across the entire router.
Info:    Install operation 5 completed successfully at 05:09:08 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router# install add /tftp://223.255.254.254/auto/tftpboot/users/user/
asr9k-upgrade-p.pie

Fri Jul  9 05:10:31.133 UTCRP/0/RP1/CPU0:Jul  9 05:10:32.156 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Info:    Install operation 6 '(admin) install add
Info:    /tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-upgrade-p.pie'
Info:    started by user 'lab'
Info:    Install operation 6 '(admin) install add
Info:    /tftp://223.255.254.254/auto/tftpboot/users/user/asr9k-upgrade-p.pie'
Info:    started by user 'lab' via CLI at 05:10:32 UTC Fri Jul 09 2010.
Info:    The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#RP/0/RP1/CPU0:
   Jul  9 05:11:55.634 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_COMPLETED_SUCCESSFULLY :
Info:    Install operation 6 completed successfully
Info:    The following package is now available to be activated:
Info:    disk0:asr9k-upgrade-p-4.0.0
Info:    The package can be activated across the entire router.
Info:    Install operation 6 completed successfully at 05:11:55 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)# install activate disk0:asr9k-mini-p-4.0.0
disk0:asr9k-upgrade-p-4.0.0 disk0:asr9k-mpls-p-4.0.0

Fri Jul  9 05:23:23.150 UTC
Install operation 7 '(admin) install activate disk0:asr9k-mini-p-4.0.0
Info:    disk0:asr9k-upgrade-p-4.0.0 disk0:asr9k-mpls-p-4.0.0'
Info:    started by user 'lab'RP/0/RP1/CPU0:Jul  9 05:23:24.161 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_STARTED :
Info:    Install operation 7 '(admin) install activate disk0:asr9k-mini-p-4.0.0
Info:    disk0:asr9k-upgrade-p-4.0.0 disk0:asr9k-mpls-p-4.0.0'
Info:    started by user 'lab' via CLI at 05:23:24 UTC Fri Jul 09 2010.\ 1% complete:
Info:    The operation can still be aborted (ctrl-c for options)
Info:    This operation will reload the following nodes in parallel:
Info:    0/RP1/CPU0 (HRP) (SDR: Owner)
Info:    0/SM0/SP (Fabric-SP) (Admin Resource)Proceed with this install operation (y/n)?
[y]|
Info:    1% complete: The operation can still be aborted (ctrl-c for options)
Info:    Install Method: Parallel Reload/ 1% complete: The operation can still be aborted
(ctrl-c for options)
Info:    The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#SP/0/SM0/SP:
   Jul  9 05:36:41.152 : insthelper[62]: %INSTALL-INSTHELPER-6-RELOAD_NODE_INFO :
Info:    As part of install operation 7 this node (0/SM0/SP) will now reload.
Info:    The changes made to software configurations will not be persistent
Info:    across system reloads. Use the command '(admin) install commit' to
Info:    make changes persistent.
Info:    Please verify that the system is consistent following the software
RP/0/RP1/CPU0:Jul  9 05:36:43.962 : instdir[235]:
%INSTALL-INSTMGR-6-INSTALL_OPERATION_COMPLETED_SUCCESSFULLY :
Info:    Install operation 7 completed successfully
Info:    change using the following commands:
Info:    show system verify
Info:    install verify packages
Info:    Install operation 7 completed successfully at 05:36:43 UTC Fri Jul 09 2010.
rebooting .......................Initializing DDR SDRAM...found 4096 MB
Initializing ECC on bank 0Initializing ECC on bank 1
Initializing ECC on bank 2
```

```
Initializing ECC on bank 3
Turning off data cache, using DDR for first time
Initializing NVRAM...Testing a portion of DDR SDRAM ...done
Reading ID EEPROMs ..........................
Initializing SQUID ...
Initializing PCI ...PCI0 device[1]: Vendor ID 0x10eePCI0 device[1]: Device ID 0x300ePCI1
device[1]:
Device ID 0x1100PCI1 device[1]: Vendor ID 0x1013PCI1 device[2]: Device ID 0x680PCI1 device[2]:

Vendor ID 0x1095PCI1 device[3]: Device ID 0x5618PCI1 device[3]: Vendor ID 0x14e4Configuring
 MPPs ...
Configuring PCMCIA slots ...System Bootstrap, Version 1.53(20090311:225342) [CRS-1 ROMMON],

Copyright (c) 1994-2009 by Cisco Systems, Inc.
Acquiring backplane mastership ... successful
Preparing for fan initialization............ ready
Setting fan speed to 4000 RPMs  successfulReading backplane EEPROM ...
Released backplane mastership ...Board type is 0x100002 (1048578)
Switch 0 initialized
Switch 0 Port fe1: link up (100Mb Full Duplex Copper)
Enabling watchdogG4(7457-NonSMP-MV64360 Rev 3) platform with 4096 MB of main memory....

CARD_RACK_NUMBER: 0    CARD_SLOT_NUMBER: 1        CPU_INSTANCE: 1
RACK_SERIAL_NUMBER: TBC08052402
MBI Validation starts ... using Control Plane Ethernet.
DEBUG : Driving up signal strength for Intel LXT971
Our MAC address is 0005.9a3e.89da
Interface link changed state to UP.
Interface link state up.
MBI validation sending request.
HIT CTRL-C to abort
MBI validation sending request.
HIT CTRL-C to abort
MBI validation sending request.
HIT CTRL-C to abort
MBI validation sending request.
HIT CTRL-C to abort
MBI validation sending request.
HIT CTRL-C to abort
No MBI confirmation received from dSCboot: booting from
bootflash:disk0/asr9k-os-mbi-4.0.0/mbiasr9k-rp.vm
..................................................................................
##################################################################################

Restricted Rights LegendUse, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph(c) of the Commercial Computer Software
 - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph(c) (1) (ii) of the Rights in Technical
 Data and Computer
Software clause at DFARS sec. 252.227-7013.
cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
Cisco IOS XR Software for the Cisco XR Router, Version 4.0.0 Copyright (c) 2010 by Cisco
Systems, Inc.
Jul 09 05:39:21.334 : Install (Node Preparation): Booting with software activated by previous
 install
operation,errno=2
RP/0/RP1/CPU0Jul  9 05:44:45.941: syslogd_helper: [89]: dsc_event_handler: Got SysMgr dSC
event : 1
RP/0/RP1/CPU0:Jul  9 05:45:11.354 : shelfmgr[306]: %PLATFORM-SHELFMGR-3-POWERDOWN_RESET :
Node 0/2/SP is powered off due to admin power off request ios con0/RP1/CPU0 is now available
Press RETURN to get started.
RP/0/RP1/CPU0:Jul  9 05:45:27.453 : instdir[216]:
```

```
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is not committed. If the system reboots then the committed
software will be used.
Use 'install commit' to commit the active software. SYSTEM CONFIGURATION IN PROCESS
The startup configuration for this device is presently loading.
This may take a few minutes. You will be notified upon completion.
Please do not attempt to reconfigure the device until this process is complete.
User Access VerificationUsername: labPassword:
RP/0/RSP0/CPU0:router# admin
Fri Jul  9 05:45:55.941 UTC
RP/0/RSP0/CPU0:router(admin)# show platform

Fri Jul  9 05:45:59.805 UTCNode            Type                 PLIM             State
       Config State
-----------------------------------------------------------------------------------------
0/2/SP          MSC(SP)             N/A              UNPOWERED        NPWR,NSHUT,MON
0/RP1/CPU0      RP(Active)          N/A              IOS XR RUN       PWR,NSHUT,MON
0/SM0/SP        FC-40G/S(SP)        N/A              MBI-RUNNING      PWR,NSHUT,MON
0/SM1/*         UNKNOWN             N/A              PRESENT          PWR,NSHUT,MON

RP/0/RP1/CPU0:ios(admin)#
RP/0/RP1/CPU0:Jul  9 05:46:08.411 : instdir_lr[217]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is not committed. If the system reboots then the committed
software will be used.
Use 'install commit' to commit the active software.
RP/0/RP1/CPU0:Jul  9 05:50:40.918 : placed[283]: LR-PLANE-READY DECLARATIONSYSTEM
CONFIGURATION COMPLETED
RP/0/RP1/CPU0:Jul  9 05:50:57.293 : ifmgr[213]: %PKT_INFRA-LINK-3-UPDOWN :
Interface MgmtEth0/RP1/CPU0/0, changed state to Down
RP/0/RP1/CPU0:Jul  9 05:50:57.313 : ifmgr[213]: %PKT_INFRA-LINK-3-UPDOWN :
Interface MgmtEth0/RP1/CPU0/0, changed state to Up
RP/0/RSP0/CPU0:router(admin)# show platform

Fri Jul  9 05:59:36.266 UTC
Node            Type                PLIM             State            Config State
-----------------------------------------------------------------------------------------
0/2/SP          MSC(SP)             N/A              UNPOWERED        NPWR,NSHUT,MON
0/RP1/CPU0      RP(Active)          N/A              IOS XR RUN       PWR,NSHUT,MON
0/SM0/SP        FC-40G/S(SP)        N/A              IOS XR RUN       PWR,NSHUT,MON
0/SM1/*         UNKNOWN             N/A              PRESENT          PWR,NSHUT,MON

RP/0/RSP0/CPU0:router(admin)# install commit

Fri Jul  9 05:59:41.851 UTC
Install operation 8 '(admin) install commit' started by user 'lab' via CLI at
05:59:43 UTC Fri Jul 09 2010./
20% complete: The operation can no longer be aborted (ctrl-c for options)-
20% complete: The operation can no longer be aborted (ctrl-c for options)\
100% complete:
The operation can no longer be aborted (ctrl-c for options)
RP/0/RP1/CPU0:Jul  9 05:59:46.402 : instdir[216]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is now the same as the committed software.
Install operation 8 completed successfully at 05:59:46 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:
asr9k-upgrade-p-4.0.0

Fri Jul  9 05:59:58.082 UTC
Install operation 9 '(admin) install deactivate disk0:asr9k-upgrade-p-4.0.0'started
   by user 'lab' via CLI at 05:59:59 UTC
Fri Jul 09 2010.
1% complete: The operation can still be aborted (ctrl-c for options)-
1% complete: The operation can still be aborted (ctrl-c for options)
```

```
Info:     Install Method: Parallel Process Restart\
1% complete: The operation can still be aborted (ctrl-c for options)
The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#
Info:     The changes made to software configurations will not be persistent
Info:     across system reloads. Use the command '(admin) install commit' to
Info:     make changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
RP/0/RP1/CPU0:Jul  9 06:01:45.662 : instdir[216]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is not committed. If the system reboots then the committed
software will be used.
Use 'install commit' to commit the active software.
Install operation 9 completed successfully at 06:01:45 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)# install commit

Fri Jul  9 06:01:53.583 UTC
Install operation 10 '(admin) install commit' started by user 'lab' via CLI at06:01:54 UTC
 Fri Jul 09 2010./
20% complete: The operation can no longer be aborted (ctrl-c for options)-
20% complete: The operation can no longer be aborted (ctrl-c for options)\
100% complete: The operation can no longer be aborted (ctrl-c for options)
RP/0/RP1/CPU0:Jul  9 06:01:57.807 : instdir[216]:
%INSTALL-INSTMGR-4-ACTIVE_SOFTWARE_COMMITTED_INFO :
The currently active software is now the same as the committed software.
Install operation 10 completed successfully at 06:01:57 UTC Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)#
RP/0/RSP0/CPU0:router(admin)#
RP/0/RSP0/CPU0:router(admin)# install remove disk0:
asr9k-upgrade-p-4.0.0

Fri Jul  9 06:04:57.676 UTC
Install operation 11 '(admin) install remove disk0:asr9k-upgrade-p-4.0.0'started
   by user 'lab' via CLI at 06:04:58 UTC
Fri Jul 09 2010./
1% complete: The operation can no longer be aborted (ctrl-c for options)
Info:     This operation will remove the following packages:
Info:         disk0:asr9k-fpd-4.0.0
Info:         disk0:asr9k-doc-4.0.0
Info:         disk0:asr9k-k9sec-4.0.0
Info:         disk0:asr9k-sbc-4.0.0
Info:         disk0:asr9k-diags-4.0.0
Info:         disk0:asr9k-mgbl-4.0.0
Info:         disk0:asr9k-mcast-4.0.0
Info:         disk0:asr9k-mpls-4.0.0
Info:         disk0:asr9k-rout-4.0.0
Info:         disk0:asr9k-fwdg-4.0.0
Info:         disk0:asr9k-lc-4.0.0
Info:         disk0:asr9k-admin-4.0.0
Info:         disk0:asr9k-upgrade-p-4.0.0-
1% complete: The operation can no longer be aborted (ctrl-c for options)
Info:     After this install remove the following install rollback point will
Info:     no longer be reachable, as the required packages will not be present:
Info:         7\
1% complete: The operation can no longer be aborted (ctrl-c for options)
Proceed with removing these packages? [confirm]|
1% complete: The operation can no longer be aborted (ctrl-c for options)
The install operation will continue asynchronously.
RP/0/RSP0/CPU0:router(admin)#SP/0/SM0/SP:Jul
    9 06:05:03.902 : envmon[117]: %PLATFORM-ENVMON-4-ALARM : MINOR_HI alarm
cleared by host__temp__Inlet0
```

```
Install operation 11 completed successfully at 06:05:33 UTC
Fri Jul 09 2010.
RP/0/RSP0/CPU0:router(admin)#
RP/0/RSP0/CPU0:router(admin)# show install act
Fri Jul  9 06:08:11.372 UTC
Secure Domain Router: Owner  Node 0/RP1/CPU0 [HRP] [SDR: Owner]
Boot Device: disk0:    Boot Image: /disk0/asr9k-os-mbi-4.0.0/mbiasr9k-rp.vm
Active Packages:       disk0:asr9k-mpls-p-4.0.0      disk0:asr9k-mini-p-4.0.0
Admin Resources:  Node 0/SM0/SP [Fabric-SP] [Admin Resource]
Boot Device: bootflash:    Boot Image: /disk0/asr9k-os-mbi-4.0.0/sp/mbiasr9k-sp.vm
Active Packages:       disk0:asr9k-mini-p-4.0.0
RP/0/RSP0/CPU0:router(admin)#
```

Related Topics

Activation and Deactivation Prerequisites, on page 184
Adding and Activating Packages, on page 198
Deactivating and Removing Cisco IOS XR Software Packages, on page 215

Deactivating and Removing Cisco IOS XR Software Packages

When a package is deactivated, it is no longer active on the router, but the package files remain on the boot disk. The package files can be reactivated later, or they can be removed from the disk.

A package is deactivated using the following methods:

- When a newer version of a package is activated, the earlier version of the package is automatically deactivated. See *Related Topics* for more information.

Note Activating a software maintenance upgrade (SMU) does not cause any earlier SMUs or the package to which the SMU applies to be automatically deactivated.

- When an earlier version of a package is activated, the newer version is deactivated automatically. See *Related Topics* for more information.
- A specific package is deactivated using the **install deactivate** command. This command turns off the package features for a card or card type.

Before you begin

The following are the restrictions when deactivating and removing Cisco IOS XR Software packages:

- A package cannot be deleted if it is part of the running or committed software of the SDR.
- A package cannot be deactivated if that package is required by another active package. When a deactivation is attempted, the system runs an automatic check to ensure that the package is not required by other active packages. The deactivation is permitted only after all compatibility checks have been passed.
- Router reloads: If the deactivation requires a router reload, a confirmation prompt appears. Use the **install deactivate** command with the **prompt-level** **none** keywords to automatically ignore any reload confirmation prompts and proceed with the package deactivation. The router reloads if required.
- Node reloads: If a software operation requires a node reload, the configuration register for that node should be set to autoboot. If the config-register for the node is not set to autoboot, then the system automatically changes the setting and the node reloads. A message describing the change is displayed.
- FPD versions must be compatible with the Cisco IOS XR software that is running on the router; if an incompatibility exists between an FPD version and the Cisco IOS XR software, the device with the

field-programmable gate array (FPGA) may not operate properly until the incompatibility is resolved. For information on FPDs, including instructions to upgrade FPD images, see the *Upgrading FPD Cisco IOS XR Software* module of *Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*.

SUMMARY STEPS

1. Connect to the console port and log in.
2. **admin**
3. **install deactivate** { **id** *add-id* | *device* **:** *package* } [**location** *node-id*] [**test**] [**pause sw-change**]
4. (Optional) **show install inactive summary**
5. (Optional) **install verify packages**
6. **exit**
7. (Optional) **show system verify start**
8. (Optional) **show system verify** [**detail** | **report**]
9. **admin**
10. (Optional) **install commit**
11. (Optional) **install remove** { **id** *add-id* | *device* **:** *package* | **inactive** } [**test**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Connect to the console port and log in.	Establishes a CLI management session with the SDR. Connect to the console port for the active DSC. For more information on console connections, see *Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*.
Step 2	Required: **admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 3	**install deactivate** { **id** *add-id* | *device* **:** *package* } [**location** *node-id*] [**test**] [**pause sw-change**] **Example:** `RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:asr9k-diags-3.7.2`	Deactivates a package on all SDRs a router. • To deactivate all packages that were added in one or more specific install add operations, or specify packages by name, use the **id** *add-id* keyword and argument. The operation ID of an install add operation is indicated in the syslog displayed during the operation and in the output of the **show install log** command. • Use the **location** *node-id* keyword and argument to deactivate the package for a specific node, if supported.

	Command or Action	Purpose
		• Use the **pause sw-change** keywords to pause the operation after preparatory checks and before the configuration is locked for the actual deactivation. This enables you to hold the operation while you perform configuration changes, and proceed with the deactivation whenever you choose. This is useful, for example, if your workflow involves configuring a router out of the network during software changes and you want to minimize the time that the router is out of the network. Follow the onscreen instructions to control the pausing and completion of the operation. **Note** Press **?** after a partial package name to display all possible matches available for deactivation. If there is only one match, press [TAB] to fill in the rest of the package name. When a package is deactivated for an SDR from administration EXEC mode, a notification message appears on the console for that SDR, with information on the impact of the deactivation.
Step 4	(Optional) **show install inactive summary** **Example:** `RP/0/RSP0/CPU0:router(admin)# show install inactive summary`	Displays the inactive packages on the router.
Step 5	(Optional) **install verify packages** **Example:** `RP/0/RSP0/CPU0:router(admin)# install verify packages`	Verifies the consistency of an installed software set with the package file from which it originated. This command can be used as a debugging tool to verify the validity of the files that constitute the packages, to determine if there are any corrupted files. This command also checks for corruptions of installation state files and MBI image files. This command is particularly useful when issued after the activation of a package or upgrading the Cisco IOS XR software to a major release. **Note** The **install verify packages** command can take up to two minutes per package to process.
Step 6	Required: **exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits administration EXEC mode and returns to EXEC mode.
Step 7	(Optional) **show system verify start** **Example:**	Starts the system status check.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# show system verify start`	
Step 8	(Optional) **show system verify** [**detail** | **report**] **Example:** `RP/0/RSP0/CPU0:router# show system verify`	Displays system status information. A variety of information is displayed including the memory and CPU usage, process status, protocol status, and other status information. Use this information to verify that the system is stable. • **detail**—Displays additional information at the card and processor level, including actual numbers. • **report**—Displays the same information as the default **show system verify** command **Note** Although most of the output should display the status "OK," some processes may show other output, such as "Warning." This does not specifically indicate a problem. Contact your Cisco technical support representative for more information on the output of this command.
Step 9	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 10	(Optional) **install commit** **Example:** `RP/0/RSP0/CPU0:router(admin)# install commit`	Commits the current set of packages so that these packages are used if the router is restarted. Packages can be removed only if the deactivation operation is committed. **Note** This command is entered in administration EXEC mode.
Step 11	(Optional) **install remove** { **id** *add-id* | *device* **:** *package* | **inactive** } [**test**] **Example:** `RP/0/RSP0/CPU0:router(admin)# install remove disk0:asr9k-diags-3.8.30`	Removes the inactive package. • Only inactive packages can be removed. • Packages can be removed only if they are deactivated from all cards in the router. • The package deactivation must be committed. • To remove a specific inactive package from a storage device, use the **install remove** command with the device**:** *package* arguments. • To remove all packages that were added in one or more specific **install add** operations, use the **id** *add-id* keyword and argument. The operation ID of an **install add** operation is indicated in the syslog displayed during the operation and in the output of the **show install log** command. If you specify packages according to operation ID, all the packages

	Command or Action	Purpose
		that were added by the specified operation must still be on the router. • To remove all inactive packages from all nodes in the system, use the **install remove** command with the **inactive** keyword.

Related Topics

Adding and Activating Packages, on page 198
Committing the Active Package Set, on page 206

Examples

In the following examples, a package is deactivated from the router. The changes are committed and the inactive package is removed from the router.

Deactivating the Package: Example

```
RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:asr9k
-diags-.7.2

Install operation 27 'install deactivate disk0:asr9k-diags-3.7.2' started by
user 'lab' at 23:29:37 UTC Sat Apr 15 2009.
The install operation will continue asynchronously.
Info:     The changes made to software configuration
Info:     across system reloads. Use the command 'admin install commit' to make
Info:     changes persistent.
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages
Install operation 27 completed successfully at 23:30:22 UTC Sat Apr 15 2009.
```

Committing the Active Software Set: Example

```
RP/0/RSP0/CPU0:router(admin)# install commit

Install operation 29 'install commit' started by user 'lab' at 23:39:21 UTC
Sat Apr 15 20090.
Install operation 29 completed successfully at 23:39:24 UTC Sat Apr 15 2009.
```

Displaying the Inactive Packages: Example

```
RP/0/RSP0/CPU0:router(admin)# show install inactive summary

Default Profile:
  SDRs:
  Owner
  Inactive Packages:
    disk0:asr9k-diags-3.7.2
```

Removing the Inactive Package from the Router: Example

The following example shows how to remove an inactive package. In this example, the operation is run in test mode. The operation is confirmed and the package is removed.

```
RP/0/RSP0/CPU0:router(admin)# install remove disk0:asr9k-diags-3.7.2 test

Install operation 30 'install remove disk0:hfr-diags-3.7.2 test' started by
user 'lab' at 23:40:22 UTC Sat Apr 15 2009.
Warning:  No changes will occur due to 'test' option being specified. The
Warning:  following is the predicted output for this install command.
Info:     This operation will remove the following package:
Info:         disk0:asr9k-diags-3.7.2
Info:     After this install remove the following install rollback points will
Info:     no longer be reachable, as the required packages will not be present:
Info:         4, 9, 10, 14, 15, 17, 18
Proceed with removing these packages? [confirm] y

The install operation will continue asynchronously.
Install operation 30 completed successfully at 23.
```

Pausing Before Configuration Lock: Example

The following example shows how to deactivate a package, pausing the operation before locking the configuration for the actual software deactivation. While the operation is paused, you can enter a configuration mode and perform configurations. When you want to complete the operation, enter the **install operation** *id* **complete** command, or the **install operation** *id* **attach synchronous** command.

```
RP/0/RSP0/CPU0:router(admin)# install deactivate disk0:comp-asr9k
-3.7.2.07I.CSCsr09575-1.0.0 pause sw-change

Install operation 12 '(admin) install deactivate
  disk0:comp-asr9k-3.7.2.07I.CSCsr09575-1.0.0 pause sw-change'
  started by user 'admin' via CLI at 09:06:26 BST Mon Jul 07 2009.
Info: This operation will reload the following nodes in parallel:
Info: 0/0/CPU0 (RP) (SDR: Owner)
Info: 0/1/CPU0 (LC(E3-GE-4)) (SDR: Owner)
Info: 0/5/CPU0 (LC(E3-OC3-POS-4)) (SDR: Owner)
Proceed with this install operation (y/n)? [y]
The install operation will continue asynchronously.
Info: Install Method: Parallel Reload
Info: Install operation 12 is pausing before the config lock is applied for
Info:    the software change as requested by the user.
Info: No further install operations will be allowed until the operation is resumed.
Info: Please continue the operation using one of the following steps:
Info: - run the command '(admin) install operation 12 complete'.
Info: - run the command '(admin) install operation 12 attach synchronous' and then
Info:      answer the query.
```

Rolling Back to a Previous Software Set

Cisco IOS XR software allows you to roll back one or more SDRs to a previous committed or uncommitted software set. Use the **show install rollback ?** command to view the available rollback points and use the **install rollback to** command to roll back the SDR to a previous software set. You can also use the **install rollback to committed** command to roll back to the most recent committed software set.

Note Rollback operations can be performed by running the command in administration EXEC or EXEC mode.

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

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

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

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

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

Displaying Rollback Points

A rollback point is created every time a software package is activated, deactivated, or committed. Use the **show install rollback ?** command to display the eligible rollback points.

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# show install rollback ?

    0  ID of the rollback point to show package information for
    2  ID of the rollback point to show package information for
```

In this example, the rollback points are 0 and 2. The rollback point with the highest number is the current software point. For example, if the last installation operation was operation 3 (activating the MPLS package) then the highest rollback point is 3, which is the same as the current software (MPLS package activated).

To easily identify specific rollback points, you can assign a label or description to a rollback point using the **install label** command.

You can enter the command in either administration EXEC mode or EXEC mode.

Displaying the Active Packages Associated with a Rollback Point

To display the active packages associated with a rollback point, use the **show install rollback** command with the *point-id* argument. This command displays the packages that are active if you roll back one or more SDRs to that installation point. For example, the **show install rollback 2** command displays the packages that are active if you roll back to rollback point 2.

```
RP/0/RSP0/CPU0:router(admin)# show install rollback 0

Tue Jun 23 06:25:06.493 PST
ID: 0, Label:
```

```
Timestamp: 23:11:20 UTC Sat Oct 28 2000

Secure Domain Router: Owner

  Node 0/RSP0/CPU0 [RP] [SDR: Owner]
    Boot Device: disk0:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/mbiasr9k-rp.vm
    Rollback Packages:
      disk0:comp-asr9k-mini-3.9.0.12I

  Node 0/1/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Rollback Packages:
      disk0:comp-asr9k-mini-3.9.0.12I

  Node 0/4/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Rollback Packages:
      disk0:comp-asr9k-mini-3.9.0.12I

  Node 0/6/CPU0 [LC] [SDR: Owner]
    Boot Device: mem:
    Boot Image: /disk0/asr9k-os-mbi-3.9.0.12I/lc/mbiasr9k-lc.vm
    Rollback Packages:
      disk0:comp-asr9k-mini-3.9.0.12I
```

You can enter the command in either administration EXEC mode or EXEC mode.

Note For more information on the command options, see the *Software Package Management Commands on Cisco IOS XR Software* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*.

Rolling Back to a Specific Rollback Point

You can roll back to a specific rollback point, including a noncommitted software set:

- If you roll back to the most recent noncommitted rollback point (with the highest number), you do not need to reload the router.
- You can repeat the rollback process one rollback point at a time without reloading if you always choose the most recent rollback point.
- If you choose a rollback point that is older than the most recent point, the impacted nodes reload, interrupting data traffic on those nodes. Before the reload occurs, you are prompted to confirm the install rollback operation.

In the following example, the system is rolled back to noncommitted rollback point 8:

```
RP/0/RSP0/CPU0:router(admin)# install rollback to 8

  Install operation 10 'install rollback to 8' started by user 'cisco' at 07:49:26
  UTC Mon Nov 14 2009.
  The install operation will continue asynchronously.
  Info:     The changes made to software configurations will not be persistent
  Info:     across system reloads. Use the command 'admin install commit' to make
  Info:     changes persistent.
```

```
Info:     Please verify that the system is consistent following the software
Info:     change using the following commands:
Info:         show system verify
Info:         install verify packages

The currently active software is the same as the committed software.

Install operation 10 completed successfully at 07:51:24 UTC Mon Nov 14 2009.
```

Rolling Back to the Last Committed Package Set

Use the **install rollback to committed** command to roll back to the last committed package set.

In the following example, the owner SDR is rolled back to the last committed package set:

```
RP/0/RSP0/CPU0:router(admin)# install rollback to committed

  Install operation 27 'install rollback to committed' started by user 'lab' at
  16:41:38 UTC Sat Nov 19 2009.
  Info:     The rollback to committed software will require a reload of impacted
  Info:     nodes because it is over multiple activation & deactivation
  Info:     operations.
  Info:     This operation will reload the following node:
  Info:         0/RP1/CPU0 (RP) (SDR: Owner)
  Info:     This operation will reload all RPs in the Owner SDR, and thereby
  Info:     indirectly cause every node in the router to reload.

  Proceed with this install operation? [confirm]

  Updating Commit Database.  Please wait...[OK]
  Info:     The changes made to software configurations will not be persistent
  Info:     across system reloads. Use the command 'admin install commit' to make
  Info:     changes persistent.
  Info:     Please verify that the system is consistent following the software
  Info:     change using the following commands:
  Info:         show system verify
  Info:         install verify packages
  Install operation 27 completed successfully at 16:42:23 UTC Sat Nov 19 2009.
```

Resetting Router to Factory Settings

The logical volumes and ROMMON variables of CPU boards on a router can be reset to factory settings using zapdisk feature. After enabling the zapdisk feature on the router, the CPU boards are reset to factory settings in the next reimage of the boards. During the reimage process, all logical volumes of the CPU boards including the files saved in harddisk: are cleaned up, and ROMMON variables of the CPU boards are reset to factory settings.

Step 1 **admin**

Example:

```
Router# admin
```

Enters the System Admin EXEC mode.

Step 2 **zapdisk set**

Example:

```
sysadmin-vm:0_RP0# zapdisk set
Fri Jul  21 22:32:29.242 UTC
result Zapdisk set command success
```

Enables zapdisk feature.

Note To disable the zapdisk feature, run the **zapdisk unset** command:

```
sysadmin-vm:0_RP0# zapdisk unset
Fri Jul  21 22:32:29.242 UTC
result Zapdisk unset command success
```

Step 3 **run**

Example:

```
sysadmin-vm:0_RP0# run
 [sysadmin-vm:0_RP0:~]$/opt/cisco/calvados/bin/nvram_dump -a
 PS1=rommon ! >
 ZAPDISK_CARD=1
```

Verifies status of zapdisk feature on the CPU board. `ZAPDISK_CARD=1` indicates that zapdisk feature is enabled; `ZAPDISK_CARD=0` indicates that zapdisk feature is disabled.

Additional References

The following sections provide references related to software package management on Cisco IOS XR software.

Related Documents

Related Topic	Document Title
Cisco IOS XR install commands	*Software Package Management Commands on the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*
Cisco IOS XR getting started material	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Cisco IOS XR master command index	*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*
Information about user groups and task IDs	*Configuring AAA Services on the Cisco ASR 9000 Series Router* module of *System Security Configuration Guide for Cisco ASR 9000 Series Routers*
ROM Monitor	*ROM Monitor Configuration Guide for Cisco ASR 9000 Routers*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 9

Upgrading Field-Programmable Devices

In general terms, *field-programmable devices* (FPDs) are hardware devices implemented on router cards that support separate software upgrades. A *field-programmable gate array* (FPGA) is a type of programmable memory device that exists on most hardware components of the router. The term *FPD* has been introduced to collectively and generically describe any type of programmable hardware device on SIPs and shared port adapters (SPAs), including FPGAs. Cisco IOS XR software provides the Cisco FPD upgrade feature to manage the upgrade of FPD images on SIPs and SPAs.

This chapter describes the information that you must know to verify image versions and to perform an upgrade for SPA or SIP FPD images when incompatibilities arise.

For complete descriptions of the FPD commands listed in this module, refer to the upcoming sections. To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 32: Feature History for Upgrading FPD Software on Cisco IOS XR Software

Release	Modification
Release 3.9.0	Support for FPD upgrades was introduced.
Release 5.3.2	Enhance FPD upgrade and downgrade behavior.
Release 6.3.1	Support for parallel FPD upgrade for power modules.

This module contains the following topics:

Upgrading Field-Programmable Device

An FPD is a field programmable logic device which contains non-volatile, re-programmable memory to define its internal wiring and functionality. The contents of this non-volatile memory are called the FPD image or FPD firmware. Over the lifespan of an FPD, FPD firmware images may need upgrades for bug fixes or functionality improvements. These upgrades are performed in the field with minimum system impact.

Prerequisites for FPD Image Upgrades

You must install the FPD pie before you install the SMUs or Service Packs. If you install the SMU or Service Packs before the FPD pie, the FPDs on the line card may not upgrade. In such cases, you must remove the SMUs and Service Packs and reload the router.

Overview of FPD Image Upgrade Support

An FPD image is used to upgrade the software on an FPD.

Whenever an image is released that supports SIPs and SPAs, a companion SIP and SPA FPD image is bundled. Generally, the FPD image is not automatically upgraded. You must manually upgrade the FPD image running on the SPA or SIP when you upgrade the Cisco IOS XR software image.

FPD versions must be compatible with the Cisco IOS XR software that is running on the router; if an incompatibility exists between an FPD version and the Cisco IOS XR software, the device with the FPGA may not operate properly until the incompatibility is resolved. An FPGA incompatibility on a SPA does not necessarily affect the running of the SPA interfaces; an FPD incompatibility on a SIP disables all interfaces for all SPAs in the SIP until the incompatibility is addressed.

Use the **show hw-module fpd** command to determine if an FPD upgrade is required. A value of 'Yes' in the Upg/Dng? (upgrade/downgrade) column indicates that an upgrade or downgrade is required.

The NCS 5500 supports upgrades for FPGA devices on its SIPs and SPAs. FPGA and ROMMON software upgrades are part of an FPD image package that corresponds to a Cisco IOS XR software image. SIPs and SPAs support manual upgrades for FPGA devices using the Cisco FPD upgrade feature that is further described in this chapter.

Note

- It is mandatory to upgrade all the required FPDs before doing a reload when you are upgrading FPDs on line cards. This is because, partial FPD component upgrades might result in booting errors (in some cases).
- You must not reload any line card or the router before all FPD image upgrades are completed successfully.

Automatic FPD Upgrade

The following conditions must be met for an Automatic FPD Upgrade to work on a system upgrade:

- FPD package installation envelope (PIE) must be installed on the router.

- FPD PIE must be activated together with the new Cisco IOS XR image.
- The **fpd auto-upgrade** command must be configured in the XR Configuration mode.

The following conditions must be met for an Automatic FPD Upgrade to work on a FRU Insertion or reload:

- The **fpd auto-upgrade** command must be configured in the XR Configuration mode.

Note Although the FPD upgrade is performed during the install operation, there is no install commit performed. Therefore, once the FPD has been upgraded, if the image is rolled back to the original version, the FPD version is not downgraded to the previous version.

Note A message is displayed when router modules cannot get upgraded during automatic FPD upgrade indicating that the FPGA is intentionally skipped during upgrade. To upgrade such FPGAs, you can use the CLI command with a particular location explicitly specified. For example, **upgrade hw-module fpd all location 0/3/1**.

Restrictions For Automatic FPD Upgrade

- Newly inserted or reloaded line cards do not reload automatically after a FPD image automatic upgrade, so you must reload the line card manually to use the new FPD image
- By default, the FPD image is not automatically upgraded. You must manually upgrade the FPD image running on the Field Replaceable Unit (FRU) when you upgrade the Cisco IOS XR software image.

  However, if you enable the **fpd auto-upgrade** command in XR Configuration mode, FPD images are automatically updated when:

  - Software upgrade is carried out.
  - Line cards are added to an existing router or reloaded.
- Automatic FPD Upgrade is not performed when:
  - A non-reload software maintenance upgrade (SMU) or PIE installation is performed, even where the FPD image version changes. Since a non-reload installation is, by definition, not supposed to reload the router, and an FPD upgrade requires a router reload, an Automatic FPD Upgrade is repressed.
- In all cases where the automatic FPD upgrade is not performed, you must perform a manual FPD upgrade using the **upgrade hw-module fpd** command.
- CFP2-DCO Optical modules do not support automatic-FPD upgrade.
- The TimingIC-A and TimingIC-B FPDs upgrades are excluded from the **fpd auto-upgrade** configuration and the user-executed **upgrade hw-module location all** command. The following ASR 9000 5th Generation ASR9000 Line Cards contain FPDs TimingIC-A and TimingIC-B:
  - A99-4HG-FLEX-SE
  - A99-4HG-FLEX-TR

- A9K-8HG-FLEX-SE
- A9K-8HG-FLEX-TR
- A9K-20HG-FLEX-SE
- A9K-20HG-FLEX-TR

To upgrade TimingIC-A and TimingIC-B FPDs, see Upgrade TimingIC-A and TimingIC-B FPDs, on page 235.

Parallel Power Module Upgrade

Power modules can now be upgraded in parallel on Cisco Routers. This feature lets you perform FPD upgrades on multiple power modules simultaneously. The newer power modules (V3) take more time to upgrade separately than their previous counterparts, which increases the total time taken to upgrade a full chassis to an unacceptable limit.

Parallel upgrade process reduces the overall time required to upgrade a full chassis with many power modules. Only power modules that support FPD upgrades can be upgraded in parallel. This includes V3 AC-DC and V2 AC-DC power modules.

Note Power module upgrades are time consuming and cannot be implicitly upgraded or as a part of automatic FPD upgrades. These modules must be upgraded independent of the other fpga upgrades.

To upgrade the power modules in parallel, use **upgrade hw-module location pm-all fpd all** or **upgrade hw-module fpd all location pm-all** command in Admin mode.

To force a power module upgrade, use **upgrade hw-module fpd all force location pm-all** command in Admin mode.

Pre-requisites to perform Parallel Upgrade

- Ensure that all power connections to the power supply are energized. To verify the power supply details, use **show environment power-supply** command in Admin mode.
- Ensure power available to the power supply is equal to the rated power. For example, 6KW power module must have a 6KW power feed. If the power feed to the power supply is less, the excess power calculation will be incorrect and the chassis may run out of power during an upgrade and suffer a sudden shutdown.
- Ensure sufficient or excess power is available in the chassis before you start the upgrade process.
- Do not add or remove any component (Line cards, RPs, power connections) from the chassis during an upgrade. This may cause power failure in the system due to sudden change in power in the system.

Note

- The system upgrades the power modules in random order.
- The number of modules that can be upgraded simultaneously depends on the excess power available to the chassis.
- Ensure you initiate the parallel upgrade process only when all the pre-requisites are satisfied because the upgrade process cannot be aborted in between.

Performing Parallel Power Module Upgrade

To initiate a parallel upgrade process and upgrade all the power modules in the chassis simultaneously, use **pm-all** keyword in the **upgrade hw-module fpd** command in Admin mode.

Example

The following section illustrates parallel power module upgrade implementation:

Verification

Use **show hw-module fpd** command to verify the upgrade:

Manual Power Module Upgrade

Manual Power modules FPD upgrades are supported on Cisco ASR 9000 Series Routers and should be performed in Admin mode only. This feature lets you perform FPD upgrades on individual Power Entry Modules (PEMs) rather than initiating a Parallel Power Module Upgrade.

Only power modules that support FPD upgrades can be upgraded manually. This includes V3 AC-DC and V2 AC-DC power modules

Note Power module upgrades are time consuming and can't be implicitly upgraded or as a part of automatic FPD upgrades. These modules must be upgraded independent of the other fpga upgrades.

To determine which PEMs requires upgrade, use **show hw-module location all fpd**.

PEMs requiring upgrade are in **UPGD SKIP** status.

```
Router#show hw-module location all fpd

Auto-upgrade:Enabled
                                                                       FPD Versions
                                                                     ==============

Location    Card type             HWver FPD device        ATR Status    Running   Programd

-----------------------------------------------------------------------------------------

0/PT0       PWR-4.4KW-DC-V3       2.11  PM0-DT-Pri0MCU        UPGD SKIP   3.00      3.00

0/PT0       PWR-4.4KW-DC-V3       2.11  PM0-DT-Pri1MCU        UPGD SKIP   3.00      3.00

0/PT0       PWR-4.4KW-DC-V3       2.11  PM0-DT-Sec054vMCU     UPGD SKIP   3.00      3.00
```

```
0/PT0       PWR-4.4KW-DC-V3       2.11  PM0-DT-Sec154vMCU    UPGD SKIP   3.00      3.00

0/PT0       PWR-4.4KW-DC-V3       2.11  PM0-DT-Sec5vMCU      UPGD SKIP   3.00      3.00
```

To upgrade the power modules manually, use [**admin**] **upgrade hw-module location 0/PT**<location> **fpd** <fpd_device>.

```
Router# admin
Router# upgrade hw-module location 0/PT0 fpd PM0-DT-Pri0MCU
```

Automatic Line Card Reload on FPD Upgrade

This feature automatically reloads a newly inserted line card (LC) after a successful FPD upgrade. The current auto FPD upgrade process does not reload the line card automatically, the user had to manually reload the LC. To enable this feature on Cisco IOS XR 32 bit operating system, use the **fpd auto-reload** command and use **fpd auto-reload enable** command in Cisco IOS XR 64 bit OS.

Implementation Considerations

The following limitation must be considered while configuring automatic line card reload on FPD upgrade:

- In Cisco IOS XR 32-bit OS, FPDs that are part of MPAs are not auto upgraded neither on inserting them to a line card nor when the entire line card gets inserted into a chassis.
- In Cisco IOS XR 64-bit OS, FPDs that are part of MPAs are auto upgraded. But the MPA will not be auto reloaded.
- If the FPD upgrade fails on a line card then the automatic line card reload feature (if enabled) stops the LC from reloading.

Configuring Automatic Line Card Reload on FPD Upgrade

The auto-reload feature works only if auto-upgrade feature is also configured on the router. The following sample shows how to configure auto-reload feature for Cisco IOS XR 32-bit OS:

```
RP/0/RSP0/CPU0:ios(config)#admin
RP/0/RSP0/CPU0:ios(admin-config)#fpd auto-upgrade
RP/0/RSP0/CPU0:ios(admin-config)#fpd auto-reload
RP/0/RSP0/CPU0:ios(admin-config)#commit
```

The auto-reload feature is only supported on line cards.

The following sample shows how to configure auto-reload feature for Cisco IOS XR 64-bit OS:

```
RP/0/RSP1/CPU0:ios# config
RP/0/RSP1/CPU0:ios(config)#fpd auto-upgrade enable
RP/0/RSP1/CPU0:ios(config)#fpd auto-reload enable
RP/0/RSP1/CPU0:ios(config)#commit
```

Note During the FPD upgrade process, the linecard may display IOS XR RUN state before triggering auto-reload.

Note **To manually reload the line card on FPD upgrade**

During FPD upgrade process, ensure to use **hw-module location** *node-id* **reload** command in EXEC or administration EXEC mode at the end of the upgrade procedure. This cause the selected card(s) to perform a complete hardware reload, which is required for some FPDs.

Hitless FPD Firmware Upgrade for Optical Transceiver Modules

Table 33: Feature History Table

Feature Name	Release Information	Feature Description
Hitless FPD Firmware Upgrade for Optical Transceiver Modules	Release 7.11.1	We now support a hitless FPD firmware upgrade for some optical transceiver modules where the FPD upgrades without shutting down the laser on the optical line card modules. This functionality avoids operational downtime. This process is more efficient and avoids traffic loss. Prior to this release, the FPD upgrades used to shut down the laser. This resulted in traffic loss. This feature is supported on the following variants: • A9K-8HG-FLEX-SE/TR • A9K-20HG-FLEX-SE/TR • A99-10X400GE-X-SE/TR • A9903-20HG-PEC

We are introducing hitless FPD firmware upgrade process to support firmware upgrade without any traffic loss.

In earlier releases, when the FPD upgrade was triggered on the linecards with the Cisco 400G QSFP-DD High-Power (Bright), QSFP-DD ZR+ and ZR optical modules, the laser in these optical modules used to shut down. Once the upgrade was successful, the laser was turned on. This process resulted in traffic loss. From Cisco IOS XR Software Release 7.11.1, as hitless FPD firmware upgrade is introduced, the process of shutting down the laser is eliminated. This makes the FPD upgrade process more robust and avoids traffic loss. This feature enabled by default. The upgrade process is applicable to manual FPD upgrades.

FPD upgrade service

The main tasks of the FPD upgrade service are:

- Check FPD image version to decide if a specific firmware image needs an upgrade or not.
- Manual FPD Image Upgrade using the **upgrade hw-module fpd** command.
- Invoke the appropriate device driver with a name of the new image to load.

An FPD image package is used to upgrade FPD images. The **install activate** command is used to place the FPD binary files into the expected location on the boot devices.

Supported Upgrade Methods

Method	Remarks
Manual Upgrade	Upgrade using CLI, force upgrade supported.

Determining Upgrade Requirement

Use the **show hw-module fpd** command to determine if an FPD upgrade is required. Check for NEED UPGD in the Status column.

Example

```
Router: #show hw - module fpd

Wed Dec 14 07:08:08.424 UTC

Auto-upgrade:Disabled
                                                                      FPD Versions
                                                                     ==============
Location   Card type            HWver FPD device       ATR Status    Running   Programd
------------------------------------------------------------------------------------------

0/0        NC55-18H18F          1.0   MIFPGA               NEED UPGD   7.01      7.01

0/0        NC55-18H18F          1.0   Bootloader           CURRENT     1.14      1.14
0/0        NC55-18H18F          1.0   IOFPGA               CURRENT     0.07      0.07
0/0        NC55-18H18F          1.0   SATA-M600-MCT        CURRENT     0.23      0.23
```

Use the **show fpd package** command to find out which FPGAs are supported with your current software release and minimum hardware requirements for each module.

Automatic FPD upgrade

Use the **fpd auto-upgrade enable** command to enable the auto upgrade feature.

The FPD images are upgraded as part of the install activation of the new image. The FPDs are upgraded before the router is reloaded.

During an FPD auto-upgrade, the installed FPD rpm package includes an FPD image with a new version of software that is different than the version of the image running on the hardware. Once the FPDs have been upgraded, even if the base image is rolled backed to the older version, the FPD will not be downgraded to its previous version.

When a reload package is installed with new FPD images, the FPD images are upgraded before the router gets reloaded. This feature is controlled through an fpd auto-upgrade configuration option. The auto-upgrade feature does not address the following:

- FPD Upgrade during initial boot
- FPD Upgrade during new card insertion

Manual FPD upgrade

Manual FPD upgrade is performed using the **upgrade hw-module fpd** command. All cards or all FPGA in a card can be upgraded. If reload is required to activate FPD, the upgrade should be complete. Line-cards, fabric cards and RP cards cannot be reloaded during the process of the FPD upgrade.

FPD upgrade is transaction-based:

- Each fpd upgrade CLI execution is one transaction.
- Only one transaction is allowed at any given time.
- One transaction may include one or many FPD upgrades

The **force** option can be used to forcibly upgrade the FPD (regardless of whether it is required or not). It triggers all FPDs to be upgraded or downgraded. The **force** option can also be used to downgrade or upgrade the FPGAs even after the version check.

Note

- Sometimes, FPDs can have primary and backup images.
- Force FPD upgrade with **upgrade hw-module location all fpd all force** command affects forwarding over BVI interface. You must reload involved locations to recover.
- The use of the **force** option when performing an FPD upgrade is not recommended except under explicit direction from Cisco engineering or TAC for a one-time purpose only.
- FPD upgrade should be performed in Admin mode only.
- A new FPD upgrade should be issued only when previous FPD upgrades have been completed on the same FPD with the following syslog message:

```
RP/0/RP0/CPU0:May 10 10:11:44.414 UTC: fpd-serv[205]: %INFRA-FPD_Manager-1-UPGRADE_ALERT
 : FPD Upgrade Completed (use "show hw-module fpd" to check upgrade status)
```

Upgrade TimingIC-A and TimingIC-B FPDs

Perform the following steps to upgrade timing IC-A and Timing IC-B FPDs:

- Upgrade Timing IC-A FPD.

```
Router#upgrade hw-module location 0/[slot-number] fpd TimingIC-A
```

- Upgrade TimingIC-B FPD.

```
Router#upgrade hw-module location 0/[slot-number] fpd TimingIC-B
```

- Run the new XR using the **install commit** command, if you're performing this manual FPD upgrade.

```
Router(admin)#install commit
```

If you don't perform the install commit of the new XR, the LC reinstalls itself with this new XR again which could take 30 minutes.

- Reload the 5th Generation ASR9000 Line Card.

```
Router#admin
sysadmin-vm:0_RP0#hw-module location 0/[slot-number] reload
```

YANG Data Model for Field Programmable Device

Table 34: Feature History Table

Feature Name	Release Information	Description
Unified Model for FPD: Cisco-IOS-XR-um-fpd-cfg	Release 7.7.1	We have introduced the Cisco-IOS-XR-um-fpd-cfg unified model to enable or disable the automatic reload and automatic upgrade of Field Programmable Devices. You can access this unified model from the Github repository.

YANG is a data modeling language that helps to create configurations, retrieve operational data and execute actions. The router acts on the data definition when these operations are requested using NETCONF RPCs. The data model handles the following types of requirements on the routers for FPD:

Operational Data	Native Data Model	CLI Commands
Auto Upgrade: Enabling or disabling of automatic upgrade of FPD.	Cisco-IOS-XR-fpd-infra-cfg.yang	• **fpd auto-upgrade enable** • **fpd auto-upgrade disable**
Auto Reload: Enabling or disabling of automatic reload of FPD.	Cisco-IOS-XR-fpd-infra-cfg.yang	• **fpd auto-reload enable** • **fpd auto-reload disable**

You can access the data models from the Github repository. To learn more about the data models and put them to use, see the *Programmability Configuration Guide for Cisco ASR 9000 Series Routers*.

How to Upgrade FPD Images

You must determine if an FPD image upgrade is needed using the **show hw-module fpd** command and perform the upgrade, if needed, under the following circumstances:

- Migrate the software to a later Cisco IOS XR software release.

- Swap line cards from a system running a different Cisco IOS XR software release.
- Insert a new line card.

In the event of an FPD incompatibility with your card, you might receive the following error message:

```
LC/0/0/CPU0:Jul 5 03:00:18.929 UTC: optics_driver[220]: %L2-OPTICS-3-BAD_FPGA_IMAGE :
Detected bad MI FPGA image programmed in MI FPGA SPI flash in 0/0/CPU0 location: Failed to
 validate meta data CRC
LC/0/0/CPU0:Jul 5 03:00:19.019 UTC: optics_driver[220]: %L2-OPTICS-3-BACKUP_FPGA_LOADED :
Detected Backup FPGA image running on 0/0/CPU0 - primary image corrupted (@0x8c = 0x44)
RP/0/RP0/CPU0:Jul 5 03:00:48.987 UTC: fpd-serv[301]: %PKT_INFRA-FM-3-FAULT_MAJOR : ALARM_MAJOR
 :FPD-NEED-UPGRADE :DECLARE :0/0:
```

Upgrades to the Cisco IOS XR software might result in an FPD incompatibility. Ensure that you perform the FPD upgrade procedure and resolve all incompatibilities, for the cards to function properly.

Note The use of the **force** option when performing a FPD upgrade is not recommended except under explicit direction from Cisco engineering or TAC for a one-time purpose only.

Before you begin

- The FPD upgrade procedure is performed while the card is online. At the end of the procedure the card must be reloaded before the FPD upgrade is complete. To reload the card, you can use the **hw-module location <location> reload** command in Admin mode, during the next maintenance window. The upgrade procedure is not complete until the card is reloaded.
- During the FPD upgrade, you *must not* do the following:
  - Reload, perform an online insertion and removal (OIR) of a line card (LC), or power down the chassis. Doing so may cause the node to enter an unusable state.
  - Press **Ctrl-C** if the console appears to hang without any output. Doing so may abort the upgrade.
- If you are not sure whether a card requires an FPD upgrade, you can install the card and use the **show hw-module fpd** command to determine if the FPD image on the card is compatible with the currently running Cisco IOS XR software release.

Procedure

	Command or Action	Purpose

Step 1	**show hw-module fpd location** {**all** | *node-id*}
	Example:
	`RP/0/RSP0/CPU0:router# show hw-module fpd location all`
	or
	`RP/0/RSP0/CPU0:router# show hw-module fpd location 0/4/cpu0`
		Displays the current FPD image versions for the specified card or all cards installed in the router. Use this command to determine if you must upgrade the FPD image on your card.

	Command or Action	Purpose
Step 2	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 3	(Optional) **show fpd package** **Example:** `RP/0/RSP0/CPU0:router(admin)# show fpd package`	Displays which cards are supported with your current Cisco IOS XR software release, which FPD image you need for each card, and what the minimum hardware requirements are for the various modules. (A minimum hardware requirement version of 0.0 indicates that all hardware can support this FPD image version.) If there are multiple FPD images for your card, use this command to determine which FPD image to use if you want to upgrade only a specific FPD type.

Step 4	**upgrade hw-module fpd** {**all** | *fpga-type*} [**force**] **location** [**all** | *node-id*]
	Example:
	`RP/0/RSP0/CPU0:router(admin)# upgrade hw-module fpd`
	`all location 0/3/1`
	`.`
	`.`
	`.`
	`Successfully upgraded 1 FPD for SPA-2XOC48POS/RPR`
	`  on location 0/3/1`
	
	`RP/0/RP0/CPU0:V3_DC_MT(admin)# upgrade hw-module fpd all location all`
	`RP/0/RP0/CPU0:May 14 22:06:38.715 : upgrade_fpd_cli[65878]: %PLATFORM-UPGRADE_FPD-6-STATUS_UPG_LOC_ALL_OPT : pm fpga11 instance 14 on location 0/RP0/CPU0 was intentionally skipped during upgrade using location all option`
		Upgrades all the current FPD images that must be upgraded on the specified card with new images.
		Before continuing to the next step, wait for confirmation that the FPD upgrade has successfully completed. Status messages, similar to these, are displayed to the screen until the FPD upgrade is completed:
		`FPD upgrade started.`
		`FPD upgrade in progress..`
		`FPD upgrade in progress..`
		`FPD upgrade sent to location xxxx`
		`FPD upgrade sent to location yyyy`
		`FPD upgrade in progress..`
		`FPD upgrade finished for location xxx`
		`FPD upgrade in progress..`
		`FPD upgrade finished for location yyyy`
		`FPD upgrade completed.`
		The “FPD upgrade in progress.” message is printed every minute. These logs are information logs, and as such, are displayed if the **logging console informational** command is configured.
		If Ctrl-C is pressed while the FPD upgrade is in progress, the following warning message is displayed:
		`FPD upgrade in progress on some hardware,`
		`aborting now is not recommended as it might`
		`cause HW programming failure and result in`
		`RMA of the hardware.`
		`Do you want to continue? [Confirm(y/n)]`
		If you confirm that you want to abort the FPD upgrade procedure, this message is displayed:
		`FPD upgrade process has been aborted, please`

	Command or Action	Purpose
		`check the status of the hardware and reissue the upgrade command if required.` **Note** • If your card supports multiple FPD images, you can use the **show fpd package** admin command to determine what specific image to upgrade in the **upgrade hw-module fpd** command. • A message is displayed when router modules cannot get upgraded during upgrade with **location all** option indicating that the FPGA is intentionally skipped during upgrade. To upgrade such FPGAs, you can use the CLI command with a particular location explicitly specified. For example, **upgrade hw-module fpd all location 0/3/1**. • It is recommended to upgrade all FPGAs on a given node using the **upgrade hw-module fpd all location** {**all** | *node-id*} command. Do not upgrade the FPGA on a node using the **upgrade hw-module fpd** *<individual-fpd>* **location** {**all** | *node-id*} as it may cause errors in booting the card.
Step 5	**exit** **Example:** `sysadmin-vm:0_RP0# exit`	
Step 6	**hw-module location**{ *node-id* | **all** } **reload**	Use the **hw-module location reload** command to reload a line card. `sysadmin-vm:0_RP0# hw-module location 0/3 reload`
Step 7	**exit**	
Step 8	**show hw-module fpd**	Verifies that the FPD image on the card has been successfully upgraded by displaying the status of all FPDs in the system.

Configuration Examples for FPD Image Upgrade

The following examples indicates the use of commands associated with the FPD image upgrade procedure.

show hw-module fpd Command Output: Example

Use the **show hw-module fpd** to display the current version of FPD images on the SPAs, SIPs and other cards installed on your router.

This command can be used to identify information about FPDs on any card. If you enter the location of a line card that is not a SPA, the output displays information about any programmable devices on that line card.

The following example shows how to display FPD compatibility for all modules in the router:

```
RP/0/RSP0/CPU0:router# ios#show hw-module fpd
Tue Jan 22 13:56:55.082 UTC
                                                               FPD Versions
                                                               =================
Location   Card type        HWver FPD device       ATR Status   Running Programd
-------------------------------------------------------------------------------
0/RP0     NCS-55A2-MOD-S    0.3   MB-MIFPGA            CURRENT    0.19    0.19
0/RP0     NCS-55A2-MOD-S    0.3   Bootloader           CURRENT    1.10    1.10
0/RP0     NCS-55A2-MOD-S    0.3   CPU-IOFPGA           CURRENT    1.18    1.18
0/RP0     NCS-55A2-MOD-S    0.3   MB-IOFPGA            CURRENT    0.18    0.18
0/PM0     NC55-1200W-ACFW   1.0   LIT-PriMCU-ACFW      NEED UPGD  2.08    2.08
0/PM1     NC55-1200W-ACFW   1.0   LIT-PriMCU-ACFW      NEED UPGD  2.08    2.08
RP/0/RP0/CPU0:ios#.
```

Note After Release 5.3.x, Upg/Dng? will display Yes only for upgrade.

The following example shows the FPD for which upgrage will be skipped.

```
RP/0/RP0/CPU0:router# show hw-module fpd location all

===================================== ==========================================
                                      Existing Field Programmable Devices
                                      ==========================================
                                        HW                       Current SW Upg/
Location     Card Type                Version Type Subtype Inst   Version   Dng?
============ ======================== ======= ==== ======= ==== =========== ====
0/SM1/SP     140G-4-S1S2S3              0.1  lc   rommonA   0     2.08       Yes
--------------------------------------------------------------------------------
                                             lc    rommon   0     2.08       Yes
--------------------------------------------------------------------------------
                                             lc    fpga1    0     6.04^      No
--------------------------------------------------------------------------------
                                             lc    fpga2    0     4.01       No
--------------------------------------------------------------------------------
NOTES:
1. ^ One or more FPD will be intentionally skipped from upgrade using CLI with option "all"
 or during "Auto fpd".
    It can be upgraded only using the "admin> upgrade hw-module fpd <fpd> location <loc>"
CLI with exact location.

RP/0/RSP1/CPU0:router# show hw-module fpd location all

Mon Jun 29 05:38:50.332 PST

===================================== ==========================================
                                      Existing Field Programmable Devices
```

```
                                      ==========================================
                                        HW                        Current SW Upg/
Location     Card Type                Version Type Subtype Inst   Version   Dng?
============ ======================== ======= ==== ======= ==== =========== ====
0/RSP0/CPU0  A9K-RSP-4G                 4.8   lc   fpga3   0       1.13     No
                                              lc   fpga1   0       1.5      No
                                              lc   fpga2   0       1.14     No
                                              lc   cbc     0       1.2      No
                                              lc   fpga4   0       1.6      No
                                              lc   rommon  0       1.0      No
--------------------------------------------------------------------------------
0/RSP0/CPU0  ASR-9010-FAN               1.0   lc   cbc     1       4.0      No
--------------------------------------------------------------------------------
0/RSP0/CPU0  ASR-9010-FAN               1.0   lc   cbc     2       4.0      No
--------------------------------------------------------------------------------
0/1/CPU0     A9K-40GE-B                 1.0   lc   fpga1   0       0.38     No
                                              lc   fpga2   0       0.8      No
                                              lc   cbc     0       2.2      No
                                              lc   cpld1   0       0.15     No
                                              lc   rommon  0       1.0      No
--------------------------------------------------------------------------------
0/1/CPU0     A9K-40GE-B                 1.0   lc   fpga1   1       0.38     No
--------------------------------------------------------------------------------
0/4/CPU0     A9K-8T/4-B                 1.0   lc   fpga1   0       0.38     No
                                              lc   fpga2   0       0.10     No
                                              lc   cbc     0       2.2      No
                                              lc   cpld2   0       0.7      No
                                              lc   cpld1   0       0.15     No
                                              lc   cpld3   0       0.3      No
                                              lc   rommon  0       1.0      No
                                              lc   fpga3   0      14.42     No
--------------------------------------------------------------------------------
0/4/CPU0     A9K-8T/4-B                 1.0   lc   fpga1   1       0.38     No
--------------------------------------------------------------------------------
0/6/CPU0     A9K-4T-B                   1.0   lc   fpga1   0       0.38     No
                                              lc   fpga2   0       0.10     No
                                              lc   cbc     0       2.2      No
                                              lc   cpld2   0       0.7      No
                                              lc   cpld1   0       0.15     No
                                              lc   cpld3   0       0.3      No
                                              lc   rommon  0       1.0      No
                                              lc   fpga3   0      14.42     No
--------------------------------------------------------------------------------
0/6/CPU0     A9K-4T-B                   1.0   lc   fpga1   1       0.38     No
--------------------------------------------------------------------------------
```

The following example shows how to display FPD compatibility for a specific module in the router:

Table 35: show hw-module fpd Field Descriptions

Field	Description
Location	Location of the module in the *rack*/*slot*/*module* notation.
Card Type	Module part number.
HW Version	Hardware model version for the module.

Field	Description
Type	Hardware type. Can be one of the following types: • spa—Shared port adapter • lc—Line card
Subtype	FPD type. Can be one of the following types: • fabldr—Fabric downloader • fpga1—Field-programmable gate array • fpga2—Field-programmable gate array 2 • fpga3—Field-programmable gate array 3 • fpga4—Field-programmable gate array 4 • fpga5—Field-programmable gate array 5 • rommonA—Read-only memory monitor A • rommon—Read-only memory monitor B
Inst	FPD instance. The FPD instance uniquely identifies an FPD and is used by the FPD process to register an FPD.
Current SW Version	Currently running FPD image version.
Upg/Dng?	Specifies whether an FPD upgrade or downgrade is required. A downgrade is required in rare cases when the version of the FPD image has a higher major revision than the version of the FPD image in the current Cisco IOS XR software package.

show fpd package Command Output: Example

Use the **show fpd package** command in administration EXECAdmin EXEC mode mode to find out which line cards are supported with your current Cisco IOS XR software release, which FPD image package you need for each line card, and what the minimum hardware requirements are for each module. If multiple FPD images are available for your card, they are listed as Subtype fpga2, fpga3, and so on.

Note The FPD name used in the FPD Description column of the output of the `show fpd package` command includes the last ten characters of DCO-PID. Depending on the slot and port numbers, the FPD name is appended with DCO_0, DCO_1, or DCO_2. For example, the FPD names for `CFP2-WDM-D-1HL` in port 0 and port 1 are `-WDM-D-1HL_DCO_0` and `WDM-D-1HL_DCO_1` respectively.

Note The FPD name used in the FPD Description column of the output of the `show fpd package` command displays QDD_*instance_port-number*. For example, depending on the instance and the port number, the FPD names for the QDD-400G-ZR-S and QDD-400G-ZRP-S modules will be QDD_0_3, QDD_1_0, and so on.

> **Note** In case of Cisco ASR 9903 routers, the output of the `show fpd package` command displays QDD_*bay_port-number*. For example, depending on the bay and the port number, the FPD names for the QDD-400G-ZR-S and QDD-400G-ZRP-S modules will be QDD_0_3, QDD_0_4, and so on.

The following example shows sample output from the **show fpd package** command:

```
show fpd package
Tue Jan 22 13:56:00.212 UTC

=============================== ================================================
                                        Field Programmable Device Package
                                ================================================
                                              Req     SW      Min Req   Min Req
Card Type            FPD Description          Reload  Ver     SW Ver   Board Ver
=================== ========================== ====== ======= ======== =========
--------------------------------------------------------------------------------
NC55-1200W-ACFW      LIT-PriMCU-ACFW(A)         NO      2.09    2.09    0.0
--------------------------------------------------------------------------------
NC55-900W-ACFW-I     LIT-PriMCU-ACFW-I(A)       NO      1.04    1.04    0.0
--------------------------------------------------------------------------------
NC55-900W-DCFW-I     LIT-PriMCU-DCFW-I(A)       NO      2.260   2.260   0.0
--------------------------------------------------------------------------------
NC55-930W-DCFW-C     LIT-PriMCU-DCFW-C(A)       NO      2.259   2.259   0.0
--------------------------------------------------------------------------------
NC55-MPA-12T-S       MPAFPGA                    YES     0.27    0.27    0.0
--------------------------------------------------------------------------------
NC55-MPA-1TH2H-S     -WDM-D-1HL_DCO_2           NO     38.518  38.518   0.1
                     MPAFPGA                    YES     0.53    0.53    0.0
                     WDM-DE-1HL_DCO_2           NO     38.518  38.518   0.1
                     WDM-DS-1HL_DCO_2           NO     38.268  38.268   0.1
--------------------------------------------------------------------------------
NC55-MPA-2TH-HX-S    -WDM-D-1HL_DCO_0           NO     38.518  38.518   0.1
                     -WDM-D-1HL_DCO_1           NO     38.518  38.518   0.1
                     MPAFPGA                    YES     0.53    0.53    0.0
                     WDM-DE-1HL_DCO_0           NO     38.518  38.518   0.1
                     WDM-DE-1HL_DCO_1           NO     38.518  38.518   0.1
                     WDM-DS-1HL_DCO_0           NO     38.268  38.268   0.1
                     WDM-DS-1HL_DCO_1           NO     38.268  38.268   0.1
--------------------------------------------------------------------------------
NC55-MPA-2TH-S       -WDM-D-1HL_DCO_0           NO     38.518  38.518   0.1
                     -WDM-D-1HL_DCO_1           NO     38.518  38.518   0.1
                     MPAFPGA                    YES     0.53    0.53    0.0
                     WDM-DE-1HL_DCO_0           NO     38.518  38.518   0.1
                     WDM-DE-1HL_DCO_1           NO     38.518  38.518   0.1
                     WDM-DS-1HL_DCO_0           NO     38.268  38.268   0.1
                     WDM-DS-1HL_DCO_1           NO     38.268  38.268   0.1
--------------------------------------------------------------------------------
NC55-MPA-4H-HD-S     MPAFPGA                    YES     0.53    0.53    0.0
--------------------------------------------------------------------------------
NC55-MPA-4H-HX-S     MPAFPGA                    YES     0.53    0.53    0.0
--------------------------------------------------------------------------------
NC55-MPA-4H-S        MPAFPGA                    YES     0.53    0.53    0.0
--------------------------------------------------------------------------------
NC55A2-MOD-SE-H-S    Bootloader(A)              YES     1.11    1.11    0.0
                     CPU-IOFPGA(A)              YES     1.18    1.18    0.1
                     MB-IOFPGA(A)               YES     0.18    0.18    0.1
                     MB-MIFPGA                  YES     0.19    0.19    0.0
                     SATA(A)                    NO      5.00    5.00    0.0
--------------------------------------------------------------------------------
NCS-55A2-MOD-HD-S    Bootloader(A)              YES     1.11    1.11    0.0
```

```
                    CPU-IOFPGA(A)            YES     1.18    1.18    0.1
                    MB-IOFPGA(A)             YES     0.18    0.18    0.1
                    MB-MIFPGA                YES     0.19    0.19    0.0
                    SATA(A)                  NO      5.00    5.00    0.0
--------------------------------------------------------------------------------
NCS-55A2-MOD-HX-S   Bootloader(A)            YES     1.11    1.11    0.0
                    CPU-IOFPGA(A)            YES     1.18    1.18    0.1
                    MB-IOFPGA(A)             YES     0.18    0.18    0.1
                    MB-MIFPGA                YES     0.19    0.19    0.0
                    SATA(A)                  NO      5.00    5.00    0.0
--------------------------------------------------------------------------------
NCS-55A2-MOD-S      Bootloader(A)            YES     1.11    1.11    0.0
                    CPU-IOFPGA(A)            YES     1.18    1.18    0.1
                    MB-IOFPGA(A)             YES     0.18    0.18    0.1
                    MB-MIFPGA                YES     0.19    0.19    0.0
                    SATA(A)                  NO      5.00    5.00    0.0
--------------------------------------------------------------------------------
NCS-55A2-MOD-SE-S   Bootloader(A)            YES     1.11    1.11    0.0
                    CPU-IOFPGA(A)            YES     1.18    1.18    0.1
                    MB-IOFPGA(A)             YES     0.18    0.18    0.1
                    MB-MIFPGA                YES     0.19    0.19    0.0
                    SATA(A)                  NO      5.00    5.00    0.0
                    STATSFPGA                YES     0.01    0.01    0.0
```

This table describes the significant fields shown in the display:

Table 36: show fpd package Field Descriptions

Field	Description
Card Type	Module part number.
FPD Description	Description of all FPD images available for the line card.
Type	Hardware type. Possible types can be: • spa—Shared port adapter • lc—Line card
Subtype	FPD subtype. These values are used in the **upgrade hw-module fpd** command to indicate a specific FPD image type to upgrade.
SW Version	FPD software version recommended for the associated module running the current Cisco IOS XR software.
Min Req SW Vers	Minimum required FPD image software version to operate the card. Version 0.0 indicates that a minimum required image was not programmed into the card.
Min Req HW Vers	Minimum required hardware version for the associated FPD image. A minimum hardware requirement of version 0.0 indicates that all hardware can support this FPD image version.

Note: In the **show fpd package** command output, the “subtype” column shows the FPDs that correspond with each line card image. To upgrade a specific FPD with the **upgrade hw-module fpd** command, replace the *fpga-type* argument with the appropriate FPD from the “subtype” column, as shown in the following example:

```
RP/0/RSP0/CPU0:router(admin)# upgrade hw-module fpd fpga2 location 0/3/1 reload
```

upgrade hw-module fpd Command Output: Example

Use the **upgrade hw-module fpd** command to upgrade the FPD image on a line card. The upgrade can be executed for all FPDs or for specific FPDs that need an upgrade. To upgrade all FPDs, use **upgrade hw-module fpd all location all** command. To upgrade a specific FPD image type, use the FPD subtype value in the **upgrade hw-module fpd** command.

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# upgrade hw-module fpd fpga location 0/1/cpu0

 Mon Jan 12 05:44:37.611 PST

% RELOAD REMINDER:  - The upgrade operation of the target module will not interrupt its
normal
    operation. However, for the changes to take effect, the target module
    will need to be manually reloaded after the upgrade operation. This can
    be accomplished with the use of "hw-module <target> reload" command.
  - If automatic reload operation is desired after the upgrade, please use
    the "reload" option at the end of the upgrade command.
  - The output of "show hw-module fpd location" command will not display
    correct version information after the upgrade if the target module is
    not reloaded.
Continue? [confirm] y

Starting the upgrade/download of following FPD:

=========== ==== ======= ======= =========== =========
                                   Current    Upg/Dng
Location    Type Subtype Upg/Dng   Version    Version
=========== ==== ======= ======= =========== =========
0/1/CPU0    lc   fpga    upg         0.40         0.40
------------------------------------------------------
LC/0/1/CPU0:Jan 12 05:44:43.700 : lc_fpd_upgrade[192]: %PLATFORM-UPGRADE_FPD-6-START :
  Starting to upgrade fpga subtype image from 0.4 to 0.4 for for this card  on location
0/1/CPU0
LC/0/1/CPU0:Jan 12 05:44:42.990 : fabricq_mgr[152]: EES:Internal clock detect IDLE
  period(-106461) more than threshold(1200000)
LC/0/1/CPU0:Jan 12 05:44:42.990 : ingressq[179]: EES:Internal clock detect IDLE
  period(-106461) more than threshold(1200000)
LC/0/1/CPU0:Jan 12 05:45:09.240 : fabricq_mgr[152]: EES:Internal clock detect IDLE
  period(-105945) more than threshold(1200000)
LC/0/1/CPU0:Jan 12 05:45:09.241 : ingressq[179]: EES:Internal clock detect IDLE
  period(-105944) more than threshold(1200000)
SP/0/1/SP:Jan 12 05:45:16.020 : upgrade_daemon[280]: ...programming...
SP/0/1/SP:Jan 12 05:45:16.034 : upgrade_daemon[280]: ...it will take a while...
SP/0/1/SP:Jan 12 05:45:16.053 : upgrade_daemon[280]: ...it will take a while...
SP/0/1/SP:Jan 12 05:47:42.967 : upgrade_daemon[280]: ...programming...
SP/0/1/SP:Jan 12 05:47:42.981 : upgrade_daemon[280]: ...it will take a while...
```

```
% SLC/0/1/CPU0:Jan 12 05:48:08.737 : lc_fpd_upgrade[192]: %PLATFORM-UPGRADE_FPD-6-PASSED :

  Successfully upgrade fpga subtype image for for this card  on location 0/1/CPU0
```

show platform Command Output: Example

Use the **show platform** command to verify that the line card is up and running.

Troubleshooting Problems with FPD Image Upgrades

This section contains information to help troubleshoot problems that can occur during the upgrade process.

Power Failure or Removal of a SPA During an FPD Image Upgrade

If the FPD upgrade operation is interrupted by a power failure or the removal of the SPA, it could corrupt the FPD image. This corruption of the FPD image file makes the SPA unusable by the router and the system displays the following messages when it tries to power up the SPA. When it cannot successfully power up the SPA, it places it in the failed state, as shown in the following example:

```
LC/0/3/CPU0:Feb  4 08:23:16.672 : spa_192_jacket[188]: %L2-SPA-5-OIR_INSERTED : SPA discovered
 in bay 0
  LC/0/3/CPU0:Feb  4 08:23:23.349 : spa_192_jacket[188]: %L2-SPA-5-OIR_ERROR : SPA (0): An
 error occurred (0x1002), error recovery action: reset SPA
  LC/0/3/CPU0:Feb  4 08:23:26.431 : spa_192_jacket[188]: %L2-SPA-5-OIR_INSERTED : SPA
discovered in bay 0
  LC/0/3/CPU0:Feb  4 08:23:32.593 : spa_192_jacket[188]: %L2-SPA-5-OIR_ERROR : SPA (0): Too
 many retries, error recovery stopped
  LC/0/3/CPU0:Feb  4 08:23:32.593 : spa_192_jacket[188]: %L2-SPA-5-OIR_ERROR : SPA (0): An
 error occurred (0x1002), error recovery action: hold SPA in reset
```

When a SPA is in the failed state, it may not register itself with the FPD upgrade mechanism. In this case, you do not see the SPA listed when you use the **show hw-module fpd** command. To verify the state of a SPA, use the **show hw-module subslot error** command and the **show hw-module subslot status** command.

Performing a SPA FPD Recovery Upgrade

To recover a SPA from the failed state because of a corrupted FPD image, you must manually shut down the SPA. Use the **hw-module subslot** *subslot-id* **shutdown** command in Global Configuration mode to administratively shutdown the SPA. After the SPA is shut down, you can use the **upgrade hw-module fpd** command in administration EXEC mode:

```
RP/0/RSP0/CPU0:router# admin
RP/0/RSP0/CPU0:router(admin)# upgrade hw-module fpd fpga location 0/3/0
```

Performing a SIP FPD Recovery Upgrade

If a SIP upgrade fails for whatever reason, do not reload the SIP. Try to perform the upgrade procedure again. You can perform the upgrade procedure multiple times, as long as you do not reload the SIP. The FPD upgrade procedure takes several minutes to complete; do not interrupt the procedure. If you reload the SIP when the FPD image is corrupted, the SIP malfunctions and you must contact Cisco technical support for assistance.

To recover a SIP from the failed state because of a corrupted FPD image, you must contact Cisco technical support.

To recover a SIP from the failed state because of a corrupted FPD image, you must turn off the automatic reset of the SIP card. Use the **hw-module reset auto disable** command in administration configuration mode, as shown in the following example:

```
RP/0/RSP0/CPU0:router(admin-config)# hw-module reset auto disable location 0/1/4
```

CHAPTER 10

Configuring Network Time Protocol

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

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

For more information about NTP on the Cisco IOS XR software and complete descriptions of the NTP commands listed in this module, see Related Documents, on page 275. To locate documentation for other commands that might appear in the course of running a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 37: Feature History for Implementing NTP on Cisco IOS XR Software

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.9.0	Support was added for IPv6 addresses, VRFs, multicast-based associations, and burst and iburst modes for poll-based associations.
Release 4.3.0	Support was added for NTP-PTP interworking.
Release 4.3.1	Support was added for NTP server inside VRF interface

This module contains the following topics:

Prerequisites for Implementing NTP on Cisco IOS XR Software

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

Information About Implementing NTP

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

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

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

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

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

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

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

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

The Cisco implementation of NTP supports two ways that a networking device can obtain NTP time information on a network:

- By polling host servers
- By listening to NTP broadcasts

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

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

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

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

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

Preventing Issues due to GPS Week Number Rollover (WNRO)

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

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

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

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

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

NTP-PTP Interworking

NTP-PTP interworking provides the ability to use PTP, as well as other valid time of day (TOD) sources such as Data over Cable Service Interface Specification (DOCSIS) Timing Interface (DTI) and global positioning system (GPS), as the time source for the operating system. Prior to the support of NTP-PTP interworking, only backplane time was supported for the operating system time.

NTP-PTP interworking also provides the means to communicate status changes between PTP and NTP processes. It also supports the unambiguous control of the operating system time and backplane time in the event of bootup, switchovers or card and process failures.

Related Topics

Configuring NTP-PTP Interworking, on page 264

How to Implement NTP

Configuring Poll-Based Associations

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

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

- Client mode
- Symmetric active mode

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

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

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

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**server** *ip-address* [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**] [**burst**] [**iburst**] **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# server 172.16.22.44 minpoll 8 maxpoll 12`	Forms a server association with another system. This step can be repeated as necessary to form associations with multiple devices.

	Command or Action	Purpose

Step 4	**peer** *ip-address* [**version** *number*] [**key** *key-id*] [**minpoll** *interval*] [**maxpoll** *interval*] [**source** *type interface-path-id*] [**prefer**]
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# peer 192.168.22.33`
	`    minpoll 8 maxpoll 12 source tengige 0/0/0/1`
		Forms a peer association with another system. This step can be repeated as necessary to form associations with multiple systems.
		Note To complete the configuration of a peer-to-peer association between the router and the remote device, the router must also be configured as a peer on the remote device.

Step 5	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before`
		`  exiting(yes/no/cancel)?`
		`[cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Broadcast-Based NTP Associates

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

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

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	(Optional) **broadcastdelay** *microseconds* **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# broadcastdelay 5000`	Adjusts the estimated round-trip delay for NTP broadcasts.
Step 4	**interface** *type interface-path-id* **Example:**	Enters NTP interface configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-ntp)# interface POS 0/1/0/0`	
Step 5	**broadcast client** **Example:** `RP/0/RSP0/CPU0:router(config-ntp-int)# broadcast client`	Configures the specified interface to receive NTP broadcast packets. **Note** Go to next step to configure the interface to send NTP broadcast packets.
Step 6	**broadcast** [**destination** *ip-address*] [**key** *key-id*] [**version** *number*] **Example:** `RP/0/RSP0/CPU0:router(config-ntp-int)# broadcast destination 10.50.32.149`	Configures the specified interface to send NTP broadcast packets. **Note** Go to previous step to configure the interface to receive NTP broadcast packets.

Step 7	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp-int)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp-int)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring NTP Access Groups

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

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

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**access-group**{**peer** | **query-only** | **serve** | **serve-only**} *access-list-name* **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# access-group peer access1`	Creates an access group and applies a basic IPv4 or IPv6 access list to it.
Step 4	Use one of the following commands: • **end** • **commit** **Example:**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before`

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		`exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring NTP Authentication

This task explains how to configure NTP authentication.

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

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

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

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

SUMMARY STEPS

1. **configure**
2. **ntp**
3. **authenticate**
4. **authentication-key** *key-number* **md5** [**clear** | **encrypted**] *key-name*
5. **trusted-key** *key-number*

6. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**authenticate** **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# authenticate`	Enables the NTP authentication feature.
Step 4	**authentication-key** *key-number* **md5** [**clear** | **encrypted**] *key-name* **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# authentication-key 42 md5 clear key1`	Defines the authentication keys. • Each key has a key number, a type, a value, and, optionally, a name. Currently the only key type supported is **md5**.
Step 5	**trusted-key** *key-number* **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# trusted-key 42`	`Defines trusted authentication keys.` • If a key is trusted, this router only synchronizes to a system that uses this key in its NTP packets.

Step 6	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

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

Disabling NTP Services on a Specific Interface

NTP services are disabled on all interfaces by default.

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	Use one of the following commands: • **no interface** *type interface-path-id*	Disables NTP services on the specified interface.

	Command or Action	Purpose

	• **interface** *type interface-path-id* **disable**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# no interface pos 0/0/0/1`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# interface POS 0/0/0/1 disable`	

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the Source IP Address for NTP Packets

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

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

SUMMARY STEPS

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

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**source** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# source POS 0/0/0/1`	Configures an interface from which the IP source address is taken. **Note** This interface is used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** keyword in the **peer** or **server** command shown in Configuring Poll-Based Associations, on page 252.

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring the System as an Authoritative NTP Server

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**master** *stratum* **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# master 9`	`Makes the router an authoritative NTP server.` **Note** Use the **master** command with caution. It is very easy to override valid time sources using this command, especially if a low stratum number is configured. Configuring multiple machines in the same network with the **master** command can cause instability in time keeping if the machines do not agree on the time.

	Command or Action	Purpose

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring NTP-PTP Interworking

Use this task to configure NTP to use PTP as the time source.

Before you begin

PTP must be supported and enabled on the router before NTP-PTP interworking can be configured. If PTP is not enabled, you receive an error message similar to the following when you try to commit the configuration:

```
RP/0/RSP0/CPU0:router(config)# ntp master primary-reference-clock
RP/0/RSP0/CPU0:router(config)# commit

% Failed to commit one or more configuration items. Please issue
'show configuration failed' from this session to view the errors

RP/0/RSP0/CPU0:router(config)# show configuration failed
[:::]
ntp
 master primary-reference-clock
!!% 'ip-ntp' detected the 'fatal' condition 'PTP is not supported on this platform'
!
end
```

Refer to the Configuring PTP, on page 384 module for more information.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**master primary-reference-clock** **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# master primary-reference-clock`	Specifies PTP to be the NTP time source.

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Updating the Hardware Clock

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**update-calendar** **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# update-calendar`	Configures the router t o update its system calendar from the software clock at periodic intervals.
Step 4	Use one of the following commands: • **end**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying the Status of the External Reference Clock

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

Note The commands can be entered in any order.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ntp associations** [**detail**] [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show ntp associations`	Displays the status of NTP associations.
Step 2	**show ntp status** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show ntp status`	Displays the status of NTP.

Examples

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

```
RP/0/RSP0/CPU0:router# show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
+~127.127.1.1      127.127.1.1       5     5  1024   37     0.0    0.00   438.3
*~172.19.69.1      172.24.114.33     3    13  1024    1     2.0   67.16     0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

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

```
RP/0/RSP0/CPU0:router# show ntp status

Clock is synchronized, stratum 4, reference is 172.19.69.1
nominal freq is 1000.0000 Hz, actual freq is 999.9988 Hz, precision is 2**26
reference time is C54C131B.9EECF6CA (07:26:19.620 UTC Mon Nov 24 2008)
clock offset is 66.3685 msec, root delay is 7.80 msec
root dispersion is 950.04 msec, peer dispersion is 3.38 msec
```

Configuration Examples for Implementing NTP

Configuring Poll-Based Associations: Example

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

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

  server 172.19.69.1
```

Configuring Broadcast-Based Associations: Example

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

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

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

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

Configuring NTP Access Groups: Example

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

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

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

Configuring NTP Authentication: Example

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

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

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

Disabling NTP on an Interface: Example

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

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

Configuring the Source IP Address for NTP Packets: Example

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

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

Configuring the System as an Authoritative NTP Server: Example

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

```
ntp
  master 6
```

Updating the Hardware Clock: Example

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

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

FQDN for NTP Server

Table 38: Feature History Table

Feature Name	Release Information	Feature Description
FQDN for NTP Server on Non-default VRF	Release 7.9.1	You can now specify a Fully Qualified Domain Name (FQDN) as the hostname for NTP server configuration over nondefault VRFs. FQDNs are easy to remember compared to numeric IP addresses. Service migration from one host to another can cause a change in IP address leading to outages. Prior releases allowed FQDN handling in only default VRFs.

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

Starting Cisco IOS XR Software Release 7.9.1 you can configure FQDN in nondefault vrf also.

Configure FQDN for NTP server

Prerequisites for configuring FQDN in a nondefault VRF

- Configuration must exist for DNS resolution over that specific VRF.
- The server must be reachable.

Configuration Example for FQDN on NTP Server on Default VRF

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

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

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

Running Configuration

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

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

Verification

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

```
Router#show ntp associations

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

Configuration Example for FQDN on NTP Server on Nondefault VRF

FQDN must be reachable from the router to configure it as an NTP server or peer. You can use the **ping** command and verify that FQDN is reachable. In the following example, time.cisco.com is the FQDN and vrf_1 is the VRF over which it is reachable.

```
Router#ping time.cisco.com vrf vrf_1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.1 timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 171/171/172 ms
```

When you have confirmed that FQDN is pingable, you can configure FQDN to be used as an NTP server/peer. The following example shows how to configure an NTP server using its FQDN over a nondefault vrf.

```
Router#configure
Router(config)#ntp server vrf vrf_1 time.cisco.com minpoll 4 maxpoll 4 iburst
Router(config)#commit
```

> **Note** If the FQDN you're trying to configure isn't reachable, the CLI treats it as invalid input.

Running Configuration

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

```
Router#show running-config ntp
 ntp
  server vrf vrf_1 192.0.2.1 minpoll 4 maxpoll 4 iburst
 !
```

Verification

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

```
Router#show ntp associations
 address         ref clock     st  when  poll reach  delay  offset    disp
~192.0.2.1 vrf vrf_1
                   173.38.201.115    2    14    16   37  179.10  13.492  16.680
* sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

Configuring NTP server inside VRF interface

This task explains how to configure NTP server inside VRF interface.

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

SUMMARY STEPS

1. **configure**
2. **ntp**
3. **vrf** *vrf-name*
4. **source** *interface-type interface-instance*
5. Use one of the following commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**ntp** **Example:** `RP/0/RSP0/CPU0:router(config)# ntp`	Enters NTP configuration mode.
Step 3	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router(config)# ntp vrf Customer_A`	Specify name of a VRF (VPN- routing and forwarding) instance to configure.
Step 4	**source** *interface-type interface-instance* **Example:** `RP/0/RSP0/CPU0:router(config)# ntp vrf Customer_A source bvi 70`	Configures an interface from which the IP source address is taken. This allows IOS-XR to respond to NTP queries on VRF interfaces, in this case the source is BVI. **Note** This interface is used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** keyword in the **peer** or **server** command shown in Configuring Poll-Based Associations, on page 252.

Step 5	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-ntp)# end`
	or
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		  • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Additional References

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

Related Documents

Related Topic	Document Title
Cisco IOS XR clock commands	*Clock Commands on the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*
Cisco IOS XR NTP commands	*NTP Commands on* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*
Information about getting started with Cisco IOS XR Software	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Cisco IOS XR master command index	*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*
Information about user groups and task IDs	*Configuring AAA Services on the Cisco ASR 9000 Series Router* module of *System Security Configuration Guide for Cisco ASR 9000 Series Routers*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

CHAPTER 11

Configuring Network Configuration Protocol

This module provides details of the Network Configuration Protocol. For relevant commands, see *System Security Command Reference for Cisco ASR 9000 Series Routers*.

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

The Network Configuration Protocol

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

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

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

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

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

Netconf Sessions and Operations

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

Characteristics of a netconf session:

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

Basic Netconf operations:

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

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

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

The Yang data model

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

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

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

Advantages of using the Yang model are:

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

Netconf and Yang

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

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

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

S. No.	Device / component	Action
3	Cisco router	Enable Netconf agent. Use the **netconf-yang agent ssh** and **ssh server netconf** command. The port can be selected. By default, it is set as 830.
4	Cisco router	Yang models are a part of the software image. The models can be retrieved from the router , using the <get-schema> operation.
5	Netconf client (application) The application can be on any standalone application or a SDN controller supporting Netconf	Installs and processes the Yang models. The client can offer a list of supported yang models; else the user will have to browse and locate the required yang file. There is a yang model file for each configuration module; for instance if the user wants to configure CDP , the relevant yang model is Cisco-IOS-XR-cdp-cfg **Note** Refer the table which lists all the supported yang models. Supported Yang Models , on page 280
5	Netconf client	Sends Netconf operation request over SSH to the router. A configuration request could include Yang-based XML data to the router. Currently, SSH is the only supported transport method.
6	Cisco router	Understands the Yang-based XML data and the network is configured accordingly (in case of configuration request from the client).
		The interactions between the client and the router happens until the network is configured as desired.

Supported Yang Models

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

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

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

Denial of Services Defence for Netconf-Yang

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

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

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

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

The relevant commands are discussed in detail, in the *System Security Command Reference for Cisco ASR 9000 Series Routers*

Dynamic Loading of Operational Yang Models

Netconf is enhanced to pre-load only the configurational yang models in memory, when it starts. The operational yang models are loaded into memory only when a request is issued. This helps reduce consumption of the RAM memory.

Enabling NETCONF over SSH

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

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

Prerequisites:

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**netconf-yang agent ssh** **Example:** `RP/0/RSP0/CPU0:router (config) # netconf agent ssh`	Enables NETCONF agent over SSH connection. After NETCONF is enabled, the Yang model in the controllcker, can configure the relevant models. **Note** The Yang models can be retrieved from the router via NETCONF <get-schema> operation.
Step 3	**ssh server netconf** [**vrf** *vrf-name* [**ipv4 access-list***ipv4 access list name*] [**ipv6 access-list** *ipv6 access list name*]] **Example:** `RP/0/RSP0/CPU0:router (config) # ssh server netconf vrf netconfvrf ipv4 access-list InternetFilter`	Brings up the netconf subsytem support with SSH server using a specified VRF of up to 32 characters. If no VRF is specified, the default VRF is used. To stop the SSH server from receiving any further connections for the specified VRF, use the **no** form of this command. Optionally ACLs for IPv4 and IPv6 can be used to restrict access to the netconf subsystem of the ssh server before the port is opened. **Note** The netconf subsystem support with SSH server can be configured for use with multiple VRFs .
Step 4	**ssh server netconf port** *port-number* **Example:** `RP/0/RSP0/CPU0:router (config) # ssh server netconf port 830`	Configures a port for the netconf ssh server. This command is optional. If no port is specified, port 830 is uses by default. **Note** 830 is the IANA-assigned TCP port for NETCONF over SSH, but it can be changed using this command.

What to do next

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

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

Examples: Netconf over SSH

This section illustrates some examples relevant to Netconf:

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

```
config
netconf-yang agent ssh
```

```
ssh server netconf vrf default
  !
!
```

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

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

Show command outputs

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

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

Additional Reference

Table 39: Related Documents

Related Topic	Document Title
Netconf-Yang	For related commands, see *System Security Command Reference for Cisco ASR 9000 Series Routers*

Table 40: Standards

Component	RFCs
YANG	6020
NETCONF	6241
NETCONF over SSH	6242

CHAPTER 12

Configuring Open Flow Agent

OpenFlow is a specification from the Open Networking Foundation (ONF) that defines a flowbased forwarding infrastructure (L2-L4 Ethernet switch model) and a standardized application programmatic interface (protocol definition) to learn capabilities, add and remove flow control entries and request statistics. OpenFlow allows a controller to direct the forwarding functions of a switch through a secure channel.

This module has details about the Open Flow Agent, relevant concepts and configurations.

Table 41: Feature History for Implementing OFACisco IOS XR Software

Release	Modification
Release 5.1.2	This feature was introduced.
Release 5.3.4	OnePK support was discontinued.

OpenFlow

Openflow is an open standard to communicate between controllers, which are running applications and network elements (such as, routers and switches).

For details regarding OpenFlow, please refer the OpenFlow chapter in the *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.

An overview of OFA

OpenFlow is a specification from the Open Networking Foundation (ONF) that defines a flowbased forwarding infrastructure (L2-L4 Ethernet switch model) and a standardized application programmatic interface (protocol definition) to learn capabilities, add and remove flow control entries and request statistics. OpenFlow allows a controller to direct the forwarding functions of a switch through a secure channel. Local device configuration is out of scope of the OpenFlow protocol. OpenFlow essentially provides a forwarding instruction set, allowing applications to directly program any-to-any routing and switching, with header field rewrite. New matches and actions can be applied to packets in arbitrary unconstrained fashion, allowing routing and switching on the new criteria. Routers and switches embed the fast packet forwarding and the high level routing decisions together into their software on the same device. With only a few exceptions based on user configuration, all routing and switching decisions are made by the built-in protocols and control plane logic that reside on the switch.

Prerequisites for OpenFlow Agent

The following prerequisites are required to use the OpenFlow agent on the platforms supporting IOS-XR:

- Special build of the Release 5.1.x software that has the OpenFlow functionality is required.
- The Enhanced Ethernet line card for the Cisco ASR 9000 Series Router is required for the OpenFlow agent feature.
- Any controller with version 1.1 or 1.3 is required (example, POX, ODL).
- The asr9k-k9sec Package Installation Envelope (PIE) must be present. The asr9k-mpls PIE is required for support on MPLS core (such as, PWHE).

Restrictions for OpenFlow Agent

- Same interface cannot be added to more than one logical open flow switch.
- No support for output as an action for layer3 openflow logical switch (such as pipeline131, 132).
- Only layer 3 interface support for netflow sampling statistics.

Advantages

The advantages with Open Flow Agent are:

- increases network scalability
- reduces network complexity
- allows greater application control

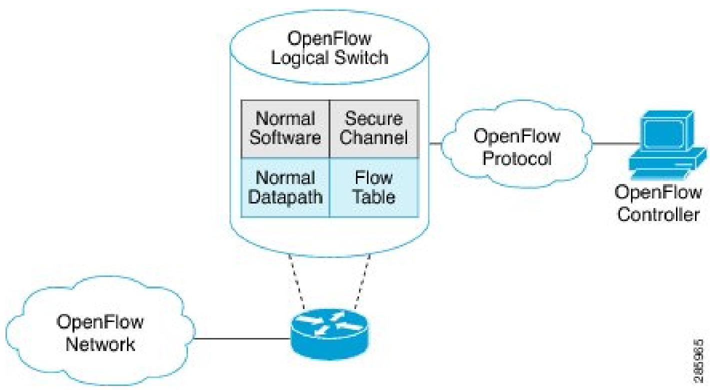
- enables customer-feature-independence

About OpenFlow

The OpenFlow protocol is based on the concept of an Ethernet switch, with an internal flow-table and standardized interface to allow traffic flows on a switch to be added or removed. The OpenFlow protocol defines the communications channel between the OpenFlow agent and the OpenFlow controller. In an OpenFlow network, the OpenFlow Agent exists on the switch and the OpenFlow controller exists on a server, which is external to the switch. Any network management is either part of the controller or accomplished through the controller.

In the Cisco OpenFlow scheme, the physical switch is divided into multiple logical switches by using the CLI to configure the connection to the controller for each logical switch and enable interfaces for each logical switch. The Openflow Agent software manages these logical switches.

The following figure shows the Cisco implementation of the OpenFlow network.



Openflow Mode for ASR9000

Openflow for the Cisco ASR 9000 Series router functions in the Integrated Hybrid mode. In this mode, both Openflow and normal switching and routing (for layer 3) operations such as L2 ethernet switching, L3 routing, etc are supported. Packets processed as the Openflow forwarding path can be processed as a normal forwarding path.

OpenFlow Table Types

An OpenFlow flow table consists of a set of flows. Each flow contains a set of matches and actions. A table has a set of capabilities in terms of supported matches and actions. Just like a policy-map, a table can be applied to a set of targets but only in the ingress direction. Hence, OpenFlow matches and actions are applied to the incoming traffic only.

Note A set of ordered tables is referred to as a pipeline. A pipeline may contain one or more ordered tables. An OpenFlow pipeline of an OpenFlow switch on ASR9K supports only one flow table.

Table 42: OpenFlow Table Types

Table Type	Pipeline	Supported Interfaces	Description
L2	129	Bridge-domain, Gigabit ethernet, Bundle, Bundle-subinterfaces, PWHE-subinterfaces	• Supports L2 header matches. • Supports L2 actions. • Can be applied to the ingress L2 interfaces.
L2_L3	130	Bridge-domain, Gigabit ethernet, Bundle, Bundle-subinterfaces, PWHE-subinterfaces	• Supports L2 and L3 (IPv4/IPv6) header matches. • Supports L2 actions. • Can be applied to the ingress L2 interfaces.
L3_V4	131	VRF and global interfaces, BVI (ipv4 only), Bridge-domain, Gigabit ethernet, Bundle, Bundle-subinterfaces	• Supports L3 (IPv4) header matches. • Supports L3 (IPv4) actions. • Can be applied to the ingress L3 interfaces.
L3_DS	132	VRF and global interfaces, BVI, Bridge-domain, Gigabit ethernet, Bundle, Bundle-subinterfaces	• Supports L2 and L3 (IPv4/IPv6) header matches. • Supports L3 (IPv4/IPv6) actions. • Can be applied to the ingress L3 interfaces.

- L2 Table--Supports L2 header matches and has L2 actions only. This table type can be applied to the ingress of an L2 interface.
- L2_L3 Table--Supports L2 and L3 header matches and has L2 actions only. Match parameters can be IPv4 or IPv6 type. This table type can be applied to the ingress of an L2 interface.
- L3_V4 Table--Supports L3 IPv4 header matches and has L3 actions only. This table type can be applied to the ingress of L3 interfaces.
- L3_DS(Dual Stack) Table--Supports L2 and L3 IPv4 and IPv6 (Dual Stack) matches and has L3 actions only. This table type can be applied to the ingress of L3 interfaces.

OpenFlow Agent Packet In and Out Feature

The Packet In and Out feature allows a flow to be programmed by the OpenFlow Agent logical switch so that packets are sent to the Controller. The special output port: **OFP_CONTROLLER** is specified for the flow action.

The Packet In and Out feature enables support for the OpenFlow output-to-port action. The output action tells the OpenFlow Agent to send all packets matching the flow to a specific port.

OpenFlow Agent with NetFlow Collection and Analytics

Applications can be provided with on-demand analytics by using the OpenFlow protocol with NetFlow. NetFlow provides statistics on packets flowing through the router, and is the standard for acquiring IP operational data from IP networks.

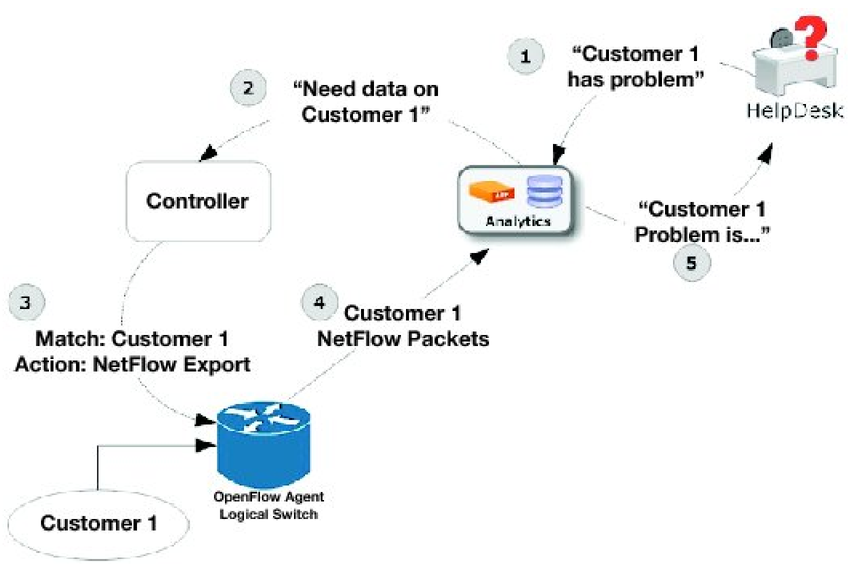
The following NetFlow maps must be configured:

- Flow Exporter Map—Specifies the destination IP address of the NetFlow collector where the NetFlow Version 9 packets are sent.
- Flow Monitor Map—Specifies the profile of the NetFlow producer, including the timeout values of active and inactive timers, size of the NetfFow cache and the exporter to be used.
- Sampler Map—Specifies how often Network Processor (NPU) needs to sample incoming and outgoing packets and create flow-packets to punt to the Line Card (LC) Central Processing Unit (CPU).

The following parameters must be specified on the OpenFlow Agent logical switch:

- Interface associated with the OpenFlow Agent logical switch that is enabled for NetFlow.
- Flow Monitor Map
- Sampler Map
- Controller IP address

Figure 10: OpenFlow Agent and NetFlow collection and analytics workflow



1. The help desk application tells the analytics application that Customer 1 has a problem.
2. The analytics application determines that it requires more information and requests more network data about Customer 1 from the Controller.
3. The Controller instructs the OpenFlow logical switch on the router to look for Customer 1 packets and generate and export NetFlow data based on Customer 1 packet flows.

4. The OpenFlow Agent logical switch exports NetFlow packets to the analytics application where they are processed.
5. The analytics application informs the help desk application of the problem.

OFA on Cisco Routers and Switches

OpenFlow SDN Applications expect network elements to speak standard OpenFlow protocol and to implement standard OpenFlow switch model. The OpenFlow Agent as a local process provides:

- OF protocol stack
- OF switch model derived from disparate Cisco software and hardware
- Version, model and feature negotiation
- Local aggregation of state and statistics
- Native dedicated CLI and troubleshooting
- High Availability

Functional Components

OpenFlow supports the configuration of multiple controllers for a logical switch. The Openflow agent can connect to a single controller or up to 8 controllers. It creates connections to all configured controllers to provide the controllers access to the OpenFlow logical switch flow tables and interfaces. It will receive flow entries from the controllers and report interface and flow status and statistics to the controllers.

The set nexthop action for layer 3 matches is implemented through a Cisco extension to the OpenFlow (1.0 and 1.3) protocol.

OFA on ASR 9000 series routers

The OpenFlow Agent supports multiple logical switch instances on ASR9K platform, with each logical switch managing a set of physical/logical interfaces, an L2 bridge domain or a VRF. Each logical switch may have one openflow connection to a single controller, or multiple connects for reliability, each to a different controller . The openflow connection to the controller uses standard TLS or plain TCP.

When the logical switch initialises a connection to the configured controller, the signaling version for the agent-controller connection is negotiated based on the bitmap version supported on both- agent and controller sides. When a logical switch starts up for the first time or at the time a logical switch loses contact with all controllers, it operates in either fail-secure mode (with default-set rule) or fail-standalone mode depending on the CLI of fail-standalone (on or off). The default for configuration is in the fail-secure mode.

OpenFlow Matches

Matches are supported on ingress port and various packet headers depending upon the packet type. Flows can have priorities. Hence, the highest priority flow entry that matches the packet gets selected.

Following table shows the list of matches supported on ASR9K for various table types:

OpenFlow Matches		OpenFlow Switch Types Supported on ASR9K			
		Applied to L2 Bridge domain		Applied to L3 or L3 VRF interface	
OXM Flow match field type for OpenFlow basic class	**Description**	**L2 only**	**L2_L3**	**L3_V4**	**L3_DS**
OFPXMT_OFB_IN_PORT	Switch input port	Yes	Yes	Yes	Yes
OFPXMT_OFB_IN_PHY_PORT	Switch physical port	No	No	No	No
OFPXMT_OFB_METADATA	Metadata passed between tables	No	No	No	No
OFPXMT_OFB_ETH_DST	Ethernet destination address	Yes	Yes	No	Yes
OFPXMT_OFB_ETH_SRC	Ethernet source address	Yes	Yes	No	Yes
OFPXMT_OFB_ETH_TYPE	Ethernet frame type	Yes	Yes	No	Yes
OFPXMT_OFB_VLAN_VID	VLAN ID	Yes	Yes	No	Yes
OFPXMT_OFB_VLAN_PCP	VLAN priority	Yes	Yes	No	Yes
OFPXMT_OFB_IP_DSCP	IP DSCP (6 bits in ToS field)	No	Yes	Yes	Yes
OFPXMT_OFB_IP_ECN	IP ECN (2 bits in ToS field)	No	No	No	No
OFPXMT_OFB_IP_PROTO	IP protocol	No	Yes	Yes	Yes
OFPXMT_OFB_IPV4_SRC	IPv4 source address	No	Yes	Yes	Yes
OFPXMT_OFB_IPV4_DST	IPv4 destination address	No	Yes	Yes	Yes
OFPXMT_OFB_TCP_SRC	TCP source port	No	Yes	Yes	Yes
OFPXMT_OFB_TCP_DST	TCP destination port	No	Yes	Yes	Yes

OpenFlow Matches		OpenFlow Switch Types Supported on ASR9K			
		Applied to L2 Bridge domain		Applied to L3 or L3 VRF interface	
OFPXMT_OFB_UDP_SRC	UDP source port	No	Yes	Yes	Yes
OFPXMT_OFB_UDP_DST	UDP destination port	No	Yes	Yes	Yes
OFPXMT_OFB_SCTP_SRC	SCTP source port	No	Yes	Yes	Yes
OFPXMT_OFB_SCTP_DST	SCTP destination port	No	No	No	No
OFPXMT_OFB_ICMPV4_TYPE	ICMP type	No	No	No	No
OFPXMT_OFB_ICMPV4_CODE	ICMP code	No	No	No	No
OFPXMT_OFB_ARP_OP	ARP opcode	No	No	No	No
OFPXMT_OFB_ARP_SPA	ARP source IPv4 address	No	No	No	No
OFPXMT_OFB_ARP_TPA	ARP target IPv4 address	No	No	No	No
OFPXMT_OFB_ARP_SHA	ARP source hardware address	No	No	No	No
OFPXMT_OFB_ARP_THA	ARP target hardware address	No	No	No	No
OFPXMT_OFB_IPV6_SRC	IPv6 source address	No	Yes	No	Yes
OFPXMT_OFB_IPV6_DST	IPv6 destination address	No	Yes	No	Yes
OFPXMT_OFB_IPV6_FLABEL	IPv6 Flow Label	No	No	No	No
OFPXMT_OFB_ICMPV6_TYPE	ICMPv6 type	No	No	No	No
OFPXMT_OFB_ICMPV6_CODE	ICMPv6 code	No	No	No	No
OFPXMT_OFB_IPV6_ND_TARGET	Target address for ND	No	No	No	No
OFPXMT_OFB_IPV6_ND_SLL	Source link-layer for ND	No	No	No	No

OpenFlow Matches		OpenFlow Switch Types Supported on ASR9K			
		Applied to L2 Bridge domain		Applied to L3 or L3 VRF interface	
OFPXMT_OFB_IPV6_ND_TLL	Target link-layer for ND	No	No	No	No
OFPXMT_OFB_MPLS_LABEL	MPLS label	No	No	No	Yes
OFPXMT_OFB_MPLS_TC	MPLS TC	No	No	No	No
OFPXMT_OFP_MPLS_BOS	MPLS BoS bit	No	No	No	Yes
OFPXMT_OFB_PBB_ISID	PBB I-SID	No	No	No	No
OFPXMT_OFB_TUNNEL_ID	Logical Port Metadata	No	No	No	No
OFPXMT_OFB_IPV6_EXTHDR	IPv6 Extension Header pseudo-field	No	No	No	No

OpenFlow Actions

Packet forwarding and packet modification types of actions are supported. The lists of actions are always immediately applied to the packet.

Note

- Only “Apply-actions” instruction (OFPIT_APPLY_ACTIONS) of OpenFlow 1.3 is supported.
- Pipeline processing instructions that allow packets to be sent to subsequent tables for further processing are not supported in this release.
- Group tables and Meter tables are not supported.

Following table shows the list of action types supported on ASR9K for various table types.

OpenFlow Actions		OpenFlow Switch Types Supported on ASR9K			
		Applied to L2 Bridge domain		Applied to L3 or L3 VRF interface	
OXM Flow action field type for OpenFlow basic class	**Description**	**L2 only**	**L2_L3**	**L3_V4**	**L3_DS**
OFPAT_OUTPUT	Output to switch port.	Yes	Yes	No	No
OFPAT_COPY_TTL_OUT	Copy TTL "outwards"	No	No	No	No

OpenFlow Actions		OpenFlow Switch Types Supported on ASR9K			
		Applied to L2 Bridge domain		Applied to L3 or L3 VRF interface	
OFPAT_COPY_TTL_IN	Copy TTL "inwards"	No	No	No	No
OFPAT_SET_MPLS_TTL	MPLS TTL	No	No	No	No
OFPAT_DEC_MPLS_TTL	Decrement MPLS TTL	No	No	No	No
OFPAT_PUSH_VLAN	Push a new VLAN tag	Yes	Yes	No	No
OFPAT_POP_VLAN	Pop the outer VLAN tag	Yes	Yes	No	No
OFPAT_PUSH_MPLS	Push a new MPLS tag	No	No	No	No
OFPAT_POP_MPLS	Pop the outer MPLS tag	No	No	No	No
OFPAT_SET_QUEUE	Set queue id when outputting to a port	No	No	No	No
OFPAT_GROUP	Apply group	No	No	No	No
OFPAT_SET_NW_TTL	IP TTL	No	No	No	No
OFPAT_DEC_NW_TTL	Decrement IP TTL	No	No	No	No
OFPAT_SET_FIELD	Set a header field using OXM TLV format	Yes	Yes	Yes	Yes
OFPAT_PUSH_PBB	Push a new PBB service tag (I-TAG)	No	No	No	No
OFPAT_POP_PBB	Pop the outer PBB service tag	No	No	No	No

Cisco Extension Actions

The set ipv4 or set ipv6 nexthop actions are used to redirect an ipv4 or ipv6 packet to the specified nexthop address, instead of using the destination address in the packet. This provides ABF (ACL Based Forwarding) kind of functionality using OpenFlow. However, VRF support and nexthop tracking as supported by CLI based ABF feature is not supported in this release.

The set fcid (Forward Class ID) action can be used to support PBTS (Policy Based Tunnel Selection) functionality using OpenFlow.

Following table shows the list of actions added by Cisco to support some extra features on ASR9K.

Cisco proprietary actions		OpenFlow Switch Types Supported on ASR9K			
		Applied to L2 Bridge domain		Applied to L3 or L3 VRF interface	
OXM Flow match field type for OpenFlow basic class	**Description**	**L2 only**	**L2_L3**	**L3_V4**	**L3_DS**
Set Ipv4 Nexthop	Set ipv4 nexthop address	No	No	Yes	Yes
Set Ipv6 Nexthop	Set ipv6 nexthop address	No	No	No	Yes
Set Forward Class ID	Set forward class ID	No	No	Yes	Yes
Set VRF	Set forward ipv4/ipv6 packet based on VRF	No	No	Yes	Yes

Set Field Actions

This table lists the set field actions supported by the Cisco ASR 9000 series router:

OpenFlow Matches		OpenFlow Switch Types Supported on ASR9K			
		Applied to L2 Bridge domain		Applied to L3 or L3 VRF interface	
OXM Flow match field type for OpenFlow basic class	**Description**	**L2 only**	**L2_L3**	**L3_V4**	**L3_DS**
OFPXMT_OFB_ETH_DST	Ethernet destination address	Yes	Yes	No	No
OFPXMT_OFB_ETH_SRC	Ethernet source address	Yes	Yes	No	No
OFPXMT_OFB_ETH_TYPE	Ethernet frame type	No	No	No	No
OFPXMT_OFB_VLAN_VID	VLAN ID	Yes	Yes	No	No
OFPXMT_OFB_VLAN_PCP	VLAN priority	Yes	Yes	No	No

OpenFlow Matches		**OpenFlow Switch Types Supported on ASR9K**			
		Applied to L2 Bridge domain		**Applied to L3 or L3 VRF interface**	
OFPXMT_OFB_IP_DSCP	IP DSCP (6 bits in ToS field)	No	No	Yes	Yes
OFPXMT_OFB_IP_ECN	IP ECN (2 bits in ToS field)	No	No	No	No
OFPXMT_OFB_IP_PROTO	IP protocol	No	No	No	No
OFPXMT_OFB_IPV4_SRC	IPv4 source address	No	No	Yes	Yes
OFPXMT_OFB_IPV4_DST	IPv4 destination address	No	No	Yes	Yes
OFPXMT_OFB_TCP_SRC	TCP source port	No	No	Yes	Yes
OFPXMT_OFB_TCP_DST	TCP destination port	No	No	Yes	Yes
OFPXMT_OFB_UDP_SRC	UDP source port	No	No	Yes	Yes
OFPXMT_OFB_UDP_DST	UDP destination port	No	No	Yes	Yes
OFPXMT_OFB_SCTP_SRC	SCTP source port	No	No	No	No
OFPXMT_OFB_SCTP_DST	SCTP destination port	No	No	No	No
OFPXMT_OFB_ICMPV4_TYPE	ICMP type	No	No	No	No
OFPXMT_OFB_ICMPV4_CODE	ICMP code	No	No	No	No
OFPXMT_OFB_ARP_OP	ARP opcode	No	No	No	No
OFPXMT_OFB_ARP_SPA	ARP source IPv4 address	No	No	No	No
OFPXMT_OFB_ARP_TPA	ARP target IPv4 address	No	No	No	No
OFPXMT_OFB_ARP_SHA	ARP source hardware address	No	No	No	No

OpenFlow Matches		OpenFlow Switch Types Supported on ASR9K			
		Applied to L2 Bridge domain		Applied to L3 or L3 VRF interface	
OFPXMT_OFB_ARP_THA	ARP target hardware address	No	No	No	No
OFPXMT_OFB_IPV6_SRC	IPv6 source address	No	No	No	No
OFPXMT_OFB_IPV6_DST	IPv6 destination address	No	No	No	No
OFPXMT_OFB_IPV6_FLABEL	IPv6 Flow Label	No	No	No	No
OFPXMT_OFB_ICMPV6_TYPE	ICMPv6 type	No	No	No	No
OFPXMT_OFB_ICMPV6_CODE	ICMPv6 code	No	No	No	No
OFPXMT_OFB_IPV6_ND_TARGET	Target address for ND	No	No	No	No
OFPXMT_OFB_IPV6_ND_SLL	Source link-layer for ND	No	No	No	No
OFPXMT_OFB_IPV6_ND_TLL	Target link-layer for ND	No	No	No	No
OFPXMT_OFB_MPLS_LABEL	MPLS label	No	No	No	No
OFPXMT_OFB_MPLS_TC	MPLS TC	No	No	No	No
OFPXMT_OFP_MPLS_BOS	MPLS BoS bit	No	No	No	No
OFPXMT_OFB_PBB_ISID	PBB I-SID	No	No	No	No
OFPXMT_OFB_TUNNEL_ID	Logical Port Metadata	No	No	No	No
OFPXMT_OFB_IPV6_EXTHDR	IPv6 Extension Header pseudo-field	No	No	No	No

Configuring OneP for Openflow

SUMMARY STEPS

1. **configure**
2. **onep**

3. **datapath transport vpathudp sender-id** *number*
4. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**onep** **Example:** `RP/0/RSP0/CPU0:router (config) # onep`	Enters the OneP configuration mode.
Step 3	**datapath transport vpathudp sender-id** *number* **Example:** `RP/0/RSP0/CPU0:router (config) # datapath transport vpathudp sender-id 1`	Configures the virtual-path udp transport datapath for the specified sender-id.
Step 4	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

Configuring a Layer 2 Logical Switch for the OpenFlow Agent

SUMMARY STEPS

1. **configure**
2. **openflow**
3. **switch** *switch -id* **pipeline** *pipeline-number*
4. **tls trust-point local** *local-tp-name* **remote** *remote-tp-name*
5. **bridge-group** *SDN-id* **bridge-domain** *switch-id*
6. **controller ipv4** *ip-address* **security [tls | none]**
7. **commit**
8. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**openflow** **Example:** `RP/0/RSP0/CPU0:router(config)# openflow`	Enters the openflow configuration mode.
Step 3	**switch** *switch -id* **pipeline** *pipeline-number* **Example:** `RP/0/RSP0/CPU0:router(config-openflow)# switch 1 pipeline 129`	Enters the logical switch configuration mode. For L2-only switch, the pipeline number is 129.
Step 4	**tls trust-point local** *local-tp-name* **remote** *remote-tp-name* **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# tls trust-point local tp1 remote tp2`	Enters the TLS configuration mode. Configures the local and remote trustpoints.
Step 5	**bridge-group** *SDN-id* **bridge-domain** *switch-id* **Example:** `RP/0/RSP0/CPU0:router (config-openflow) # bridge-group SDN-1 bridge-domain of2`	Configures the bridge-domain for the openflow switch. For layer2, the bridge-domain can be configured in the openflow switch and the interfaces of the bridge-domain will be learnt by the openflow switch.
Step 6	**controller ipv4** *ip-address* **security [tls | none]** **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# controller ipv4 5.0.1.1 port 6633 security tls`	Configures the Openflow controller for the logical switch. Configures the Openflow controller for the logical switch. Once the **controller** command is entered, a connection to the OpenFlow controller is started for the logical switch. The **tls** keyword enables the TLS connection, whereas the **none** keyword enables the TCP connection. **Note** The OpenFlow Agent can connect to a single Controller or up to 8 Controllers. Repeat this step if you need to configure additional Controllers. An openflow switch can communicate to multiple controllers (the support for high-availability is a controller functionality).
Step 7	**commit** **Example:** `RP/0/RSP0/CPU0:router(logical-switch)# commit`	Adds the Layer 2 logical switch configuration for the OpenFlow agent to the running configuration.
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

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

What to do next

Repeat these steps to configure another logical switch for the OpenFlow Agent.

Configuring a Layer 2_Layer 3 Logical Switch for the OpenFlow Agent

SUMMARY STEPS

1. **configure**
2. **openflow**
3. **switch** *switch -id* **pipeline** *pipeline-number*
4. **tls trust-point local** *local-tp-name* **remote** *remote-tp-name*
5. **bridge-group** *SDN-id* **bridge-domain** *switch-id*
6. **controller ipv4** *ip-address* **security [tls \| none]**
7. **commit**
8. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**openflow** **Example:** `RP/0/RSP0/CPU0:router(config)# openflow`	Enters the openflow configuration mode.
Step 3	**switch** *switch -id* **pipeline** *pipeline-number* **Example:** `RP/0/RSP0/CPU0:router(config-openflow)# switch 1 pipeline 130`	Enters the logical switch configuration mode. For L2_L3 switch, the pipeline number is 130.

	Command or Action	Purpose
Step 4	**tls trust-point local** *local-tp-name* **remote** *remote-tp-name* **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# tls trust-point local tp1 remote tp2`	Enters the TLS configuration mode. Configures the local and remote trustpoints.
Step 5	**bridge-group** *SDN-id* **bridge-domain** *switch-id* **Example:** `RP/0/RSP0/CPU0:router (config-openflow) # bridge-group SDN-1 bridge-domain of2`	Configures a bridge-domain for the openflow switch.
Step 6	**controller ipv4** *ip-address* **security [tls | none]** **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# controller ipv4 5.0.1.1 port 6633 security tls`	Configures the Openflow controller for the logical switch. Configures the Openflow controller for the logical switch. Once the **controller** command is entered, a connection to the OpenFlow controller is started for the logical switch. The **tls** keyword enables the TLS connection, whereas the **none** keyword enables the TCP connection. **Note** The OpenFlow Agent can connect to a single Controller or up to 8 Controllers. Repeat this step if you need to configure additional Controllers. An openflow switch can communicate to multiple controllers (the support for high-availability is a controller functionality).
Step 7	**commit** **Example:** `RP/0/RSP0/CPU0:router(logical-switch)# commit`	Adds the Layer 2 logical switch configuration for the OpenFlow agent to the running configuration.
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

What to do next

Repeat these steps to configure another logical switch for the OpenFlow Agent.

Configuring a Layer 3_VRF Logical Switch for the OpenFlow Agent

SUMMARY STEPS

1. **configure**
2. **openflow**
3. **switch** *switch -id* **pipeline** *pipeline-number*
4. **vrf IPv4**
5. **tls trust-point local** *local-tp-name* **remote** *remote-tp-name*
6. **controller ipv4** *ip-address* **security [tls | none]**
7. **commit**
8. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**openflow** **Example:** `RP/0/RSP0/CPU0:router(config)# openflow`	Enters the openflow configuration mode.
Step 3	**switch** *switch -id* **pipeline** *pipeline-number* **Example:** `RP/0/RSP0/CPU0:router(config-openflow)# switch 1 pipeline 131`	Enters the logical switch configuration mode. For L3_V4(VRF) switch, the pipeline number is 131.
Step 4	**vrf IPv4** **Example:** `RP/0/RSP0/CPU0:router(config)# vrf IPv4`	VRF configuration. All the interfaces belonging to IPv4 VRF will be learnt by the openflow switch.
Step 5	**tls trust-point local** *local-tp-name* **remote** *remote-tp-name* **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# tls trust-point local tp1 remote tp2`	Enters the TLS configuration mode. Configures the local and remote trustpoints.
Step 6	**controller ipv4** *ip-address* **security [tls | none]** **Example:**	Configures the Openflow controller for the logical switch. Configures the Openflow controller for the logical switch. Once the **controller** command is entered, a connection to the OpenFlow controller is started for the logical switch.

	Command or Action	Purpose
	RP/0/RSP0/CPU0:router(config-openflow-switch)# **controller ipv4 5.0.1.1 port 6633 security tls**	**Note** The OpenFlow Agent can connect to a single Controller or up to 8 Controllers. Repeat this step if you need to configure additional Controllers.
Step 7	**commit** **Example:** RP/0/RSP0/CPU0:router(logical-switch)# commit	Adds the Layer 2 logical switch configuration for the OpenFlow agent to the running configuration.
Step 8	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

What to do next

Repeat these steps to configure another logical switch for the OpenFlow Agent.

Configuring a Layer 3_Dual-stack Logical Switch for the OpenFlow Agent

SUMMARY STEPS

1. **configure**
2. **openflow**
3. **switch** *switch -id* **pipeline** *pipeline-number*
4. **interface** *type interface-path-id*
5. **tls trust-point local** *local-tp-name* **remote** *remote-tp-name*
6. **bridge-group** *SDN-id* **bridge-domain** *switch-id*
7. **controller ipv4** *ip-address* **security [tls | none]**
8. **commit**
9. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**openflow** **Example:** `RP/0/RSP0/CPU0:router(config)# openflow`	Enters the openflow configuration mode.
Step 3	**switch** *switch -id* **pipeline** *pipeline-number* **Example:** `RP/0/RSP0/CPU0:router(config-openflow)# switch 1 pipeline 132`	Enters the logical switch configuration mode. For L3_DS switch, the pipeline number is 132.
Step 4	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-openflow)# interface Bundle-Ether2.1`	Interface configuration. **Note** VRFs can be configured here. Both IPv4 abd IPv6 VRFs are supported.
Step 5	**tls trust-point local** *local-tp-name* **remote** *remote-tp-name* **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# tls trust-point local tp1 remote tp2`	Enters the TLS configuration mode. Configures the local and remote trustpoints.
Step 6	**bridge-group** *SDN-id* **bridge-domain** *switch-id* **Example:** `RP/0/RSP0/CPU0:router (config-openflow) # bridge-group SDN-1 bridge-domain of2`	
Step 7	**controller ipv4** *ip-address* **security [tls | none]** **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# controller ipv4 5.0.1.1 port 6633 security tls`	Configures the Openflow controller for the logical switch. Configures the Openflow controller for the logical switch. Once the **controller** command is entered, a connection to the OpenFlow controller is started for the logical switch. **Note** The OpenFlow Agent can connect to a single Controller or up to 8 Controllers. Repeat this step if you need to configure additional Controllers.
Step 8	**commit** **Example:** `RP/0/RSP0/CPU0:router(logical-switch)# commit`	Adds the Layer 2 logical switch configuration for the OpenFlow agent to the running configuration.
Step 9	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions:

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

What to do next

Repeat these steps to configure another logical switch for the OpenFlow Agent.

Enabling TLS

SUMMARY STEPS

1. **configure**
2. **openflow switch** *logical-switch-id*
3. **tls trust-point local** *local-tp-name* **remote** *remote-tp-name*
4. **commit**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**openflow switch** *logical-switch-id* **Example:** `RP/0/RSP0/CPU0:router(config)# openflow switch 100`	Enters the OpenFlow logical switch configuration mode.
Step 3	**tls trust-point local** *local-tp-name* **remote** *remote-tp-name* **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# tls trust-point local tp1 remote tp2`	Enters the TLS configuration mode. Configures the local and remote trustpoints.
Step 4	**commit** **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# commit`	Adds the logical switch configuration for the OpenFlow agent to the running configuration.

	Command or Action	Purpose
Step 5	**end** **Example:** `RP/0/RSP0/CPU0:router(config-openflow-switch)# end`	Exits logical switch configuration mode and enters EXEC mode.

Configuring NetFlow for the OpenFlow Agent

SUMMARY STEPS

1. **configure**
2. **flow exporter-map** *fem-name*
3. **destination** *location*
4. **version v9**
5. **commit**
6. **exit**
7. **flow monitor-map** *map-name*
8. **record ipv4**
9. **exporter** *map-name*
10. **cache entries** *number*
11. **cache timeout** {**active** *timeout-value* | **inactive** *timeout-value* | **update** *timeout-value*}
12. **commit**
13. **exit**
14. **sampler-map** *map-name*
15. **random 1 out-of** *sampling-interval*
16. **commit**
17. **exit**
18. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**flow exporter-map** *fem-name* **Example:** `RP/0/RSP0/CPU0:router(config)# flow exporter-map fem`	Enters flow exporter map configuration mode. **Note** A single flow monitor map can support up to eight exporters.

	Command or Action	Purpose
Step 3	**destination** *location* **Example:** `RP/0/RSP0/CPU0:router(config-fem)# destination 10.0.1.2`	Configures the export destination for the flow exporter map. The destination **location** argument can be a hostname or an IP address.
Step 4	**version v9** **Example:** `RP/0/RSP0/CPU0:router(config-fem)# version v9`	Specifies export version parameters and enters the flow exporter map version configuration mode.
Step 5	**commit** **Example:** `RP/0/RSP0/CPU0:router(config-fem-ver)# commit`	Commits the configuration changes to running to the running configuration.
Step 6	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-fem-ver)# exit`	Exits flow exporter map version configuration mode and enters global configuration mode.
Step 7	**flow monitor-map** *map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# flow monitor-map mmap`	Creates a monitor map and configures a monitor map name and enters flow monitor map configuration mode
Step 8	**record ipv4** **Example:** `RP/0/RSP0/CPU0:router(config-fmm)# record ipv4`	Configures the flow record map name for IPv4. By default, the originating autonomous system (AS) numbers are collected and exported.
Step 9	**exporter** *map-name* **Example:** `RP/0/RSP0/CPU0:router(config-fmm)# exporter fmap`	Associates an exporter map with a monitor map. **Note** A single flow monitor map can support up to eight exporters.
Step 10	**cache entries** *number* **Example:** `RP/0/RSP0/CPU0:router(config-fmm)# cache entries 4096`	(Optional) Configures the number of entries in the flow cache. Replace the number argument with the number of flow entries allowed in the flow cache, in the range from 4096 through 1000000. The default number of cache entries is 65535.
Step 11	**cache timeout** {**active** *timeout-value* | **inactive** *timeout-value* | **update** *timeout-value*} **Example:** `RP/0/RSP0/CPU0:router(config-fmm)# cache timeout active 10`	(Optional) Configures the active, inactive, or update flow cache timeout value. • The default timeout value for the inactive flow cache is 15 seconds. • The default timeout value for the active flow cache is 1800 seconds. • The default timeout value for the update flow cache is 1800 seconds.

	Command or Action	Purpose
		Note The **update** keyword and *timeout-value* argument are used for permanent caches only. It specifies the timeout value that is used to export entries from permanent caches. In this case, the entries are exported but remain the cache.
Step 12	**commit** **Example:** `RP/0/RSP0/CPU0:router(config-fmm)# commit`	Commits the configuration changes to running to the running configuration.
Step 13	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-fmm)# exit`	Exits flow monitor map version configuration mode and enters global configuration mode.
Step 14	**sampler-map** *map-name* **Example:** `RP/0/RSP0/CPU0:router(config)# sampler-map`	Creates a sampler map and enters sampler map configuration mode. **Note** When configuring a sampler map, be aware that NetFlow supports policing at a rate of 35,000 packets per second per direction for each individual line card.
Step 15	**random 1 out-of** *sampling-interval* **Example:** `RP/0/RSP0/CPU0:router(config-sm)# random 1 out-of 65535`	Configures the sampling interval to use random mode for sampling packets. For the *sampling-interval* argument, specify a number from 1 to 65535.
Step 16	**commit** **Example:** `RP/0/RSP0/CPU0:router(config-sm)# commit`	Commits the configuration changes to running to the running configuration.
Step 17	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-sm)# exit`	Exits sampler map version configuration mode and enters global configuration mode.
Step 18	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.

What to do next

Go to the “Associating the OpenFlow Agent Logical Switch with NetFlow” section to complete the second part of this configuration.

Configuration Examples: Openflow

Attaching a bridge domain to an Openflow Switch: Examples

- Attaching a L2-only Openflow switch

```
openflow
switch 1 pipeline 129
  tls trust-point local tp1 remote tp1
  bridge-group SDN-2 bridge-domain OF-2
  controller ipv4 5.0.1.200 port 6653 security tls
```

- Attaching a L2_L3 Openflow switch

```
openflow
switch 1 pipeline 130
  tls trust-point local tp1 remote tp1
  bridge-group SDN-2 bridge-domain OF-2
  controller ipv4 5.0.1.200 port 6653 security tls
```

- L3_V4 switch can be attached either to a VRF or directly to layer 3 interfaces under global VRF. In case of VRF, all the interfaces in that VRF become part of the OpenFlow switch.

```
openflow
switch 11 pipeline 131
  vrf IPv4
  controller ipv4 5.0.1.200 port 6653 security none
 !
```

- L3_DS switch can be attached either to a VRF or directly to layer 3 interfaces under global VRF.

```
openflow
switch 12 pipeline 132
  vrf IPv4
  controller ipv4 5.0.1.200 port 6653 security none
 !
```

OpenFlow Agent with NetFlow Collection and Analytics Configuration: Example

The following example describes the NetFlow exporter map configuration for the OpenFlow logical switch.

```
Device> enable
Device# configure terminal
Device(config)# flow exporter-map fem
Device(config-fem)# destination 10.0.1.2
Device(config-fem)# version v9
```

```
Device(config-fem-ver)# commit
Device(config-fem-ver)# exit
```

The following example describes the NetFlow monitor map configuration for the OpenFlow logical switch.

```
Device(config)# flow monitor-map mmap
Device(config-fmm)# record ipv4
Device(config-fmm)# exporter fmap
Device(config-fmm)# cache entries 4096
Device(config-fmm)# commit
Device(config-fmm)# exit
```

The following example describes the NetFlow sampler map configuration for the OpenFlow logical switch.

```
Device(config)# sampler-map
Device(config-sm)# random 1 out-of 65535
Device(config-sm)# commit
Device(config-sm)# exit
```

The following example describes how the OpenFlow Agent logical switch is configured so that the NetFlow collection and analytics are associated with it.

```
Device(config)# openflow switch 100 netflow
Device(logical-switch)# flow monitor mmap sampler smap
Device(logical-switch)# interface GigabitEthernet0/1/0/6
Router(logical-switch)# controller 10.0.1.2 port 6633
Device(logical-switch)# commit
Device(logical-switch)# end
```

The following example describes **show** command output for an OpenFlow Agent logical switch that is configured with NetFlow collection and analytics.

```
Device# show openflow switch 100
Fri Jan 25 14:29:21.078 UTC

Logical Switch Context
         Id:                    100
         Switch type:           Netflow
         Layer:                 NONE
         Signal version:        Openflow 1.0
         Data plane:            secure
         Fallback:              normal
         Config state:          no-shutdown
         Working state:         enabled
         TLS version:           NONE
         TLS private key:       none:none
         TLS private key file:  NONE
         TLS certificate file:  NONE
         Controller:            10.0.1.2:6633, last alive ping: 2013-01-25 14:29:20
         Netflow Monitor:       mmap
         Netflow Sampler:       smap
         Loopback i/f:          <none>
         Loopback addr:         <none>
         Interfaces:
                GigabitEthernet0/1/0/6

Device# show openflow switch 100 flows
```

```
Fri Jan 25 14:29:24.787 UTC

Logical Openflow Switch [100]:
NXST_FLOW reply (xid=0x0):
cookie=0x0, duration=204.729s, table=0, n_packets=0, n_bytes=0, priority=500 actions=netflow


Switch flow count: 1

Device# show openflow switch 100 controllers
Fri Jan 25 14:29:28.660 UTC

Logical Openflow Switch [100]:
        Controller [tcp:10.0.1.2:6633]
           role               :  Other
           connected          :  Yes
           state              :  ACTIVE
           sec_since_connect  :  487
```

Usecase for Layer2

The Scenario: Enterprise Data Center needs to perform data backup to multiple other backup sites based on the Traffic flow. The Main DC is in Vlan 100 and Backup sites are at VLAN 1000,1001,1002. These Sites are interconnected through L2VPN.

The Solution: Openflow, we can match any Layer 2 header field (in this example we have taken priority bits) and steer the traffic to go on any L2 interconnect and also rewrite the VLANs appropriately.

Usecase for Layer3

The Scenario: Three different flows from 3 different sites connected to PE1 are trying to send 350 mbps of traffic each to PE2. The bandwidth of the shortest link, Path-2 (between PE1 and PE2) is only 1 Gigabit. Hence Path-2 gets congested as soon as the third site begins to send traffic.

The Solution: Openflow controller can be used to install rules on PE1:

- Match on Flow 1 (destined to Video server) and redirect traffic to Path-2
- Match on Flow 2 (destined to Web server) and redirect traffic to Path-1
- Match on Flow 3 (destined to File transfer server) and redirect traffic to Path-3

The Inference: Effectively utilizing the network bandwidth by redirecting destination specific traffic using OpenFlow rules.

CHAPTER 13

Configuring Data Collection Manager

This module describes the configuring of the Data Collection Manager feature.

Table 43: Feature History for Configuring Data Collection Manager

Release	Modification
Release 5.2.2	This feature was introduced

Data Collection Manager

Cisco Data Collection Manager (DCM) is an efficient and reliable data collection agent that is embedded in managed devices, such as routers and switches. DCM works on a push model, which is based on a subscribe-and-notify data pattern, as opposed to the pull model, which is based on a request-and-response data pattern. The Data Collection Manager (DCM) supports advanced on-board data processing that includes baseline calculation, summary calculation, statistical distribution, and percentile computation.

Data Collection Manager and Bulkstat

The Data Collection Manager (DCM) and the bulkstat module are the vital units of a framework which enables the bulk collection mechanism to include multiple data sources and multiple data export mechanisms.

The Bulkstat client application is implemented using the DCM core services to retrieve data and export it to the user. The Bulkstat client provides the only available user interface for DCM access. The client also provides CLI access through a new set of configuration commands and MIB access through the CISCO-DATA-COLLECTION-MIB.

DCM provides data subscription service for different data sources (such as, SNMP MIB objects and show command outputs).It also provides data retrieval management and data filtering services. With DCM, one source can be allocated for periodically collecting all management data.

Bulkstat, is an application which will use DCM to provide the following:

- Profiles and data-groups for different data-sources.
- Data processing – Summary, Distribution, Percentile and Auto-baseline.
- Data exports – File.

- Calendar scheduling.

Benefits of DCM

DCM is very useful for Data Retrieval and Export and Performance Management solutions. This list includes all the benefits of DCM.

- Data export and retrieval: The Data Collection Manager (DCM) provides data retrieval management to ensure that the data collection does not impact device resources. The DCM can export data in a file format using multiple export protocols such as FTP, TFTP, Secure copy protocol (SCP), and Secure File Transfer Protocol (SFTP). The DCM provides a query mechanism with which data can be selectively exported based on the configured time interval and other selection criteria. The DCM application also provides data filtering services and exports the filtered data. You can also set primary and secondary destinations for exporting the collected data in a raw or processed format. Snapshots of the collected data can be stored for later retrieval.
- Performance Management: The Data Collection Manager (DCM) can be used to manage various aspects of performance management. It can collect data with a high granularity to help the Network Management Server (NMS) make dynamic traffic engineering decisions. DCM can also be used to collect resource variables that are important for effective capacity trend information, such as memory, queue depth, broadcast volume, buffer, Frame Relay congestion notification, and backplane utilization.
- Troubleshooting: The streaming function of the DCM can be used for real-time troubleshooting.
- SLA: A service level agreement (SLA) includes a what-if analysis for network changes and application changes, a trend for defined performance variables, exception management for defined capacity and performance variables, and QoS management. The DCM can be used to collect periodic data for reporting purposes.

Bulkstat

Two challenges that network providers usually face are data gathering and data analysis. Network providers need to gather large volumes of data to analyze the performance of the network and to have operational control over their network. Large service providers are strengthening their data gathering and analysis infrastructure. Traditionally, Simple Network Management Protocol (SNMP) agents are used to expose management data on managed systems. But, SNMP is not well suited for gathering large volumes of data, especially over short time intervals. For example, service providers charge customers depending on the network usage. Also this data must be available on customer request. Accounting applications based on SNMP polling models consume significant network bandwidth because they poll large volumes of data frequently. The SNMP protocol data unit (PDU) is a complex data type specific to SNMP and is expensive to process because the SNMP objects and tables must be sorted in a lexicographic order. All the entries in SNMP MIB tables are lexicographically ordered by their object identifiers, because there is an implied ordering in the MIB based on the order of the object identifiers. In such cases, the need to continuously poll large or bulk SNMP statistics can be avoided by using applications known as collectors to retrieve data.

The Bulkstat application is one such collector that uses the services of the Data Collection Manager (DCM) to provide the following functions:

- Collecting SNMP MIB object values.
- Processing the collected data to create summary, percentiles, and auto-baselined values.
- Exporting collected data through simple file transfers.

- Scheduling calendar events for data collection and export.

The Bulkstat application provides command-line access through a set of new configuration commands and exclusive MIB access through CISCO-DATA-COLLECTION-MIB to collect SNMP data.

You can configure Bulkstat for the following functions:

- Specify the way Bulkstat retrieves bulk statistics.
- Specify the time interval in seconds at which Bulkstat transfers data to receivers.
- Specify the maximum size of the bulk statistics file.
- Specify the context, instance, and period at which the system retrieves bulk statistics.
- Configure file-related parameters.
- Configure the interface type on which you want to collect statistics.
- View the parameters that Bulkstat uses to collect statistics by using the show bulkstat commands.

Bulkstat Configuration Elements

The following list shows the elements that you can configure using the Bulkstat interface:

- Data set
- Instance set
- Filter set
- Data group
- Process set
- Data profile
- Calendar Scheduling

Data Set

This section describes the data set elements that you can configure to collect Simple Network Management Protocol (SNMP) data and CLI data. Only objects having the same index elements can be grouped in a single object list.

The SNMP data set contains the following fields:

Name	Description	Configuration Status
Objects	Specifies the object to be collected. Multiple objects can be configured to form a data set. The textual name of the object can be used for configuring an object. If the device does not recognize the textual name, the object identifier (OID) format can be used for configuring the name.	Mandatory

Name	Description	Configuration Status
Object Alias	Specifies the optional alias name that each object can have.	Optional

The CLI data set contains the following fields:

Name	Description	Configuration Status
CLI	Specifies the CLI command for which the show output needs to be collected. More than one CLI can be specified in the same data set.	Mandatory

Filter Set

This section describes the filter configuration per object.

The filter set elements that you can configure to collect Simple Network Management Protocol (SNMP) data are described here. More than one filter of the same type can be added to the set.

Name	Description	Status
Object match	Specifies the value to be used to match against the value retrieved for the object during collection. The value provided needs to match the type of the object. If there is an error in the type matching, the configuration is not accepted. More than one value can be specified for an object, and more than one object can have matching values.	Optional

Instance Set

This section specifies the instance set elements that you can configure to collect Simple Network Management Protocol (SNMP) data. More than one instance of the same type can be added to the set. Combinations of types of instance set elements are not supported.

The SNMP Instance set contains the following fields:

Name	Description	Configuration Status
Exact	Specifies the instance for which the data should be collected. More than one instance can be specified, but only fully qualified instances should be specified.	Optional

Name	Description	Configuration Status
Wildcard	Specifies all instances for all objects under the object configured in the data set.	Optional
Range	Specifies the start and end instances. All instances within the range, including the start and end, are collected, but only fully qualified instances should be specified.	Optional
Repetition	Specifies the start of the repetition and the number of repetitions. All instances from the start until the number of repetitions within the subtree are collected.	Optional
Interface	Specifies the interface instead of the index. The ifIndex assigned to the interface will be used as an index. This can be used for MIB objects indexed by ifindex.	Optional

Process Set

Data processing allows users to derive information from raw SNMP data, by calculating summaries and percentiles. Service providers rely on monitored SNMP data to alert network management systems (NMSs) of changing network conditions. By periodically monitoring the device data and comparing it against a set of thresholds, the network can automatically alert the operators, thereby allowing efficient operations.

- Summary: You can enable summary processing on the collected object value and calculate minimum, maximum, and average values. A summary is calculated for only those objects that are marked as process capable in the data group and uses the absolute or delta value as per the object configuration.
- Distribution: You can enable distribution processing on the collected object value by specifying the object type, minimum value, maximum value, and the number of buckets to distribute the value. Based on the configuration, counters are maintained per bucket and are incremented whenever the data falls into a bucket range.
- Percentile: You can enable percentile processing on the collected object value. A percentile is calculated on every process interval expiry. Distribution configuration is mandatory to enable percentile processing. Percentile computation is done assuming that the distribution is normal.
- Auto-baseline: You can enable baseline processing on the collected object value. The baseline internally uses all summary, distribution, and percentile calculations to provide baseline values. You can configure either baseline processing or other forms of processing, such as summary, distribution, and percentile calculations. The auto-baseline feature in DCM calculates the baseline values for variables of interest on the device and allows network management applications or network operators to retrieve the baseline values. The baseline values can be displayed in terms of percentiles or a median with standard deviation.

Data Group

This section describes the data group, which contains the data-group name, data-group type, data set, instance set, filter set, polling interval, SNMP context, and other processing options.

The Data Group elements are:

Name	Description	Configuration Status
Data	Specifies any one of the data types as defined in the topic **Data Set**.	Mandatory
Instance	Specifies any one of the instance types as defined in the topic **Instance Set.**	Optional, if not specified.Default behavior of the instance set is wildcard. Only applicable for SNMP.
Filter	Specifies any one of the filter types as defined in the topic **Filter Set**.	Optional, if not specified. Only applicable for SNMP.
Polling Interval	Specifies the collection periodic interval in seconds. In case of recurring collection, the data is collected at the expiration of the collection interval until the collection is stopped.	Optional
Context	Specifies the management context from which to obtain data for this data group.	Optional
Process Summary	Enables summary processing of the data marked to be processed in the corresponding data-set configuration.	Optional
Process Distribution	Enables distribution processing of the data marked to be processed in the corresponding data-set configuration.	Optional
Process Percentile	Enables percentile processing of the data marked to be processed in the corresponding data-set configuration.	Optional

Name	Description	Configuration Status
Process Auto-baseline	Enables auto-baselining processing of the data marked to be processed in the corresponding data-set configuration. If auto-baseline process is enabled, the other processes, such as summary, distribution, and percentile configurations, if done previously, are removed because auto-baseline process uses these functionalities internally. **Note** Removing this configuration will not reinstate the other configurations that are removed.	Optional
Discard raw	Specifies whether to store raw data. If data is processed, the user can choose to store only process data by setting the option.	Optional

Data Profile

This section describes the data profile that is used to group multiple data groups. This is done to simplify the configuration and to aggregate data of similar nature. A data profile can have multiple data groups. A data group can have constraints in the data specified in the element. If two sets of data need to be written to the same file, the respective data groups should be linked as part of a single profile.

The Data Profile has these fields:

Name	Description	Status
Data groups	Specifies the data group to be linked to this profile. Multiple data groups can be linked to a single profile.	Mandatory before activating a profile
Transfer Interval	Specifies the transfer periodic interval in seconds. In case of recurring transfer, the data is transferred when the transfer interval expires.	Optional

Name	Description	Status
Process Interval	Specifies the process periodic interval in seconds. The data is processed during every collection interval as soon as it is collected. When the process interval expires, the processed data is written into a file and transferred.	Optional
Primary URL	Specifies the URL of the primary management station. The files containing the collected data are transferred to this URL when the transfer interval expires.	Mandatory
Secondary URL	Specifies the URL of the secondary management station to be used in case the transfer to the primary management station fails.	Optional
Schema	Specifies the file data format. The schema ASCII option is supported.	Optional
Retry	Specifies the number of times that the transfer is retried in case of transfer failures to both primary and secondary management stations. This command has an effect only if the retain command is configured in the profile. The retry interval is computed by dividing the retention time by the number of retries. For example, if the file is retained for 60 minutes and the retry is 6 times, the transfer is attempted every 10 minutes, until the transfer succeeds or the file is removed.	Optional
Buffer-size	Specifies the maximum size to which the file containing the collected data can grow. When it reaches the limit, the file is closed and the transfer is attempted based on the transfer configuration associated with the data group or profile.	Optional
Retention Memory	Specifies the time, in seconds, to retain the file in the memory.	Optional

Name	Description	Status
Retention USB	Specifies the time, in seconds, to retain the file in the USB. This option is available only if the device supports the USB drive.	Optional

Calendar Scheduling

The Bulkstat application allows you to schedule each subscription for collection. A subscription can be scheduled for one-time collection or periodic collection. A periodic subscription can be repeated infinitely or for a specified number of repetitions. A timer is instantiated for every activated subscription.

The calendar scheduling elements are:

Name	Description	Configuration Status
One shot	Specifies that the data is collected for a specified collection interval.	Optional
Recurring	Specifies that the data is collected regularly at the specified time, day, month, and for a specified collection interval.	Optional

File Data Export

The file data export feature on the Data Collection Manager (DCM) exports the collected data based on the transfer configurations. Data can be exported in various formats, and Bulkstat files are one such format to collect data. The format in which the data is inserted into the file conforms to the schema-Ascii format described in CISCO-DATA-COLLECTION-MIB and CISCO-BULK-FILE-MIB. The data sequence in which the data is stored is determined based on the sequence in which the data is received.

The Cisco File Transfer module is responsible for transferring the files as per the transfer configuration. A file can be retained in the device whether the transfer was a success or a failure.

Configuring an SNMP Bulkstat Data Set

The first step in configuring the Simple Network Management Protocol (SNMP) periodic data collection and transfer mechanism is to configure one or more data sets. A data set is used to group objects of similar types, based on the data source. The data set is defined outside of the data group. This external definition gives the user the flexibility to use the same data set across multiple data groups and to collect the output for different instances and different contexts.

All objects in an SNMP data set must be indexed by the same MIB index. However, the objects in the data set must not belong to the same MIB or the MIB table.

Perform this task to configure the SNMP Bulkstat data set.

SUMMARY STEPS

1. **configure**
2. **bulkstat data** *data-set -name* **type snmp**

3. **object** *oid* [**alias** *alias-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bulkstat data** *data-set -name* **type snmp** **Example:** `RP/0/RSP0/CPU0:router (config) # bulkstat data interface-stats type snmp`	Defines an SNMP Bulkstat data set and enters SNMP bulk statistics data set configuration mode. The creation of an SNMP Bulkstat data set creates a row in the cdcDGBaseObjectEntry table in the SNMP MIB.
Step 3	**object** *oid* [**alias** *alias-name*] **Example:** `RP/0/RSP0/CPU0:router (config-bs-ds-snmp) # object 1.3.6.1.2.1.2.2.1.10 alias ifInOctets`	Adds a MIB object to the SNMP Bulkstat data set. If the object is already present in the data set, this command replaces the old object configuration with the new configuration. **Note** Repeat this command until all objects to be monitored are added to this list.

Configuring an SNMP Bulkstat Filter Set

The Simple Network Management Protocol (SNMP) filter set specifies the filter configuration for every SNMP object.

Perform this task to configure the SNMP Bulkstat filter set.

SUMMARY STEPS

1. **configure**
2. **bulkstat filter** *filter-set -name*
3. **match** *object-name* { **eq** *line* | **start** *line* | **not** { **eq** *line* | **start** *line* } }

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bulkstat filter** *filter-set -name* **Example:** `RP/0/RSP0/CPU0:router (config) # bulkstat filter ifType`	Defines an SNMP Bulkstat filter set and enters SNMP bulk statistics filter set configuration mode.

	Command or Action	Purpose
Step 3	**match** *object-name* { **eq** *line* | **start** *line* | **not** { **eq** *line* | **start** *line* } } **Example:** `RP/0/RSP0/CPU0:router (config-bs-fs) # match ifType eq 6767`	(Optional) Specifies a value to be used to match against the value retrieved for the object during collection. **Note** More than one value can be specified for an object, and more than one object can have match values.

Configuring an SNMP Bulkstat Instance Set

The Simple Network Management Protocol (SNMP) instance set specifies the instances for which the data should be collected. Each subscription can collect different entries for specified objects based on the instance configuration. While more than one instance of the same type can be added to the instance set, a combination of different types is not supported.

Perform this task to configure the SNMP Bulkstat instance set.

SUMMARY STEPS

1. **configure**
2. **bulkstat instance** *instance-set -name* **type snmp**
3. **exact oid** *oid*
4. **exact interface** *interface-id*
5. **wildcard**
6. **wildcard oid** *oid*
7. **wildcard interface** *interface-id*
8. **repetition oid** *oid* **max** *value*
9. **range start** *oid* **end** *oid*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bulkstat instance** *instance-set -name* **type snmp** **Example:** `RP/0/RSP0/CPU0:router (config) # bulkstat instance exact type snmp`	Defines an SNMP Bulkstat instance set and enters SNMP Bulkstat instance set configuration mode. The creation of an SNMP Bulkstat instance set creates a row in the cdcDGInstanceEntry table in the SNMP MIB. **Note** An instance created using this command can be linked to more than one data group.
Step 3	**exact oid** *oid* **Example:**	(Optional) Indicates that the specified instance, when appended to the object list, is the complete OID.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router (config-bs-is-snmp) # exact oid 1`	
Step 4	**exact interface** *interface-id* **Example:** `RP/0/RSP0/CPU0:router (config-bs-is-snmp) # exact interface Ethernet0/0 sub-if`	(Optional) Specifies an interface name and number, for example interface Ethernet 0, instead of specifying the ifIndex OID for the interface.
Step 5	**wildcard** **Example:** `RP/0/RSP0/CPU0:router (config-bs-is-snmp) # wildcard`	(Optional) Specifies whether an object used for evaluating an expression should be made a wildcard during an event configuration.
Step 6	**wildcard oid** *oid* **Example:** `RP/0/RSP0/CPU0:router (config-bs-is-snmp) # wildcard oid 1`	(Optional) Indicates that all subindices of the specified OID belong to this schema.
Step 7	**wildcard interface** *interface-id* **Example:** `RP/0/RSP0/CPU0:router (config-bs-is-snmp) # wildcard interface Ethernet0/0 sub-if`	(Optional) Specifies an interface name and number, for example interface Ethernet 0, instead of specifying the ifIndex OID for the interface.
Step 8	**repetition oid** *oid* **max** *value* **Example:** `RP/0/RSP0/CPU0:router (config-bs-is-snmp) # repetition oid 1.2.3.4 max 2000`	(Optional) Configures data collection to repeat get-next for the maximum number of instances starting from the specified oid instance.
Step 9	**range start** *oid* **end** *oid* **Example:** `RP/0/RSP0/CPU0:router (config-bs-is-snmp) # range start 1.2.3.4 end 1.2.3.6`	(Optional) Configures a range of instances for which the data is collected.

Configuring a Bulkstat Data Group

The Bulkstat data group element is used to group the data set, filter set, and instance set and also to specify the processing options.

Perform this task to configure the Bulkstat data group.

SUMMARY STEPS

1. **configure**
2. **bulkstat data-gorup** *data-group-name*
3. **collect type** { { **command** \| **expression** } **date** *date-set-name* **filter** *filter-set-name* \| **snmp** { **data** *data-set-name* **instance** *instance-set-name* **filter** *filter-set-name* } }
4. **context** *context-name*

5. **interval polling** *polling-interval*
6. **discard**
7. **process**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bulkstat data-gorup** *data-group-name* **Example:** `RP/0/RSP0/CPU0:router (config) # bulkstat data-group if-dg`	Defines a Bulkstat data group and enters Bulkstat data group configuration mode. The creation of a Simple Network Management Protocol (SNMP) Bulkstat data group creates a row in the cdcDGEntry table in the SNMP MIB.
Step 3	**collect type** { { **command** | **expression** } **date** *date-set-name* **filter** *filter-set-name* | **snmp** { **data** *data-set-name* **instance** *instance-set-name* **filter** *filter-set-name* } } **Example:** `RP/0/RSP0/CPU0:router (config-bs-dg) # collect type snmp data interface-stats instance ins-exact filter ifType`	Specifies the collection type to collect data from different sources for this data group.
Step 4	**context** *context-name* **Example:** `RP/0/RSP0/CPU0:router (config-bs-dg) # context ctx-name`	Specifies the management context from which to obtain data for this data group.
Step 5	**interval polling** *polling-interval* **Example:** `RP/0/RSP0/CPU0:router (config-bs-dg) # interval polling 100`	Specifies the collection periodic interval in seconds. In case of recurring collection, the data is collected at the expiration of the collection interval until the collection is stopped.
Step 6	**discard** **Example:** `RP/0/RSP0/CPU0:router (config-bs-dg) # discard`	Specifies whether to discard the raw data.
Step 7	**process** **Example:** `RP/0/RSP0/CPU0:router (config-bs-dg) # process`	Configures process-related parameters for a data group.

Configuring a Bulkstat Profile

Perform this task to configure the Bulkstat Profile.

The profile element is used to group multiple data groups. This grouping simplifies the configuration and aggregates data of a similar nature. If two sets of data need to be written to the same file, the respective data groups should be linked as part of a single profile.

SUMMARY STEPS

1. **configure**
2. **bulkstat profile** *profile-name*
3. **data-group** *data-group name*
4. **interval transfer** { **process** | **raw**} *seconds*
5. **file-format schema ASCII**
6. **file retain** { **disk** *url* | **memory** *seconds* }
7. **file size** *bytes*
8. **file transfer** { **retry** *number* | **url** { **primary** *url* | **secondary** *url* } }
9. **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bulkstat profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router (config) # bulkstat profile if-stats`	Creates a profile with the given name and enters Bulkstat profile configuration mode. If the profile is already created, this command sets the context for the existing profile.
Step 3	**data-group** *data-group name* **Example:** `RP/0/RSP0/CPU0:router (config-bs-profile) # data-group if-dg`	Specifies the data group to be linked to this profile. Multiple data groups can be linked to a single profile.
Step 4	**interval transfer** { **process** | **raw**} *seconds* **Example:** `RP/0/RSP0/CPU0:router (config-bs-profile) # interval transfer process 2000`	Specifies the transfer periodic interval in seconds. In case of recurring transfer, the data is transferred at the expiration of the transfer interval until the transfer is stopped.
Step 5	**file-format schema ASCII** **Example:** `RP/0/RSP0/CPU0:router (config-bs-profile) # file-format schemaASCII`	Configures the file-related parameter for a profile. Specifies the file data format in ASCII.
Step 6	**file retain** { **disk** *url* | **memory** *seconds* }	Configures the file-related parameter for a profile.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router (config-bs-profile) # file retain memory 1500`	• **disk** - retains the file in the specified location in the disk for a specified amount of time in seconds. • **memory** - retains the file in the memory for a specified amount of time in seconds.
Step 7	**file size** *bytes* **Example:** `RP/0/RSP0/CPU0:router (config-bs-profile) # file size 2048`	Configures the file-related size parameter for a profile. **size** - Specifies the maximum buffer size in bytes. When the limit is reached, the file is closed and transfer is attempted based on the transfer configuration associated with the data group or the profile.
Step 8	**file transfer** { **retry** *number* | **url** { **primary** *url* | **secondary** *url* } } **Example:** `RP/0/RSP0/CPU0:router (config-bs-profile) # file transfer url primary tftp://20.1.1.1/iox`	Configures the file-related transfer parameter for a profile. • **primary** - specifies the URL of the primary management station. The files containing the collected data are transferred to this URL when the transfer interval expires. • **secondary** - specifies the URL to be used in case the transfer to the primary management station fails.
Step 9	**enable** **Example:** `RP/0/RSP0/CPU0:router (config-bs-profile) # enable`	Enables the profile for collection and transfer.

Configuring Bulkstat Calendar Scheduling

SUMMARY STEPS

1. **configure**
2. **bulkstat schedule** *schedule* **at** *time-detail* { **oneshot** | **recurring** }
3. **profile** *profile-name* **start** { **oneshot** | **recurring** *number* }
4. **profile** *profile-name* **stop**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**bulkstat schedule** *schedule* **at** *time-detail* { **oneshot** | **recurring** }	Defines the Bulkstat calendar scheduler set and enters Bulkstat event scheduler configuration mode.

	Command or Action	Purpose
	Example: `RP/0/RSP0/CPU0:router (config) # bulkstat schedule event1 at 11:30 jan 10 oneshot`	For the time-detail option, enter the details of the time as prompted. First the time in the 24-hour clock format, followed by the month and then the date.
Step 3	**profile** *profile-name* **start** { **oneshot** | **recurring** *number* } **Example:** `RP/0/RSP0/CPU0:router (config-bs-schedule) # profile cpu-process start recurring 5`	Creates a profile and sets the condition to enable the profile for a one-time event or enables the profile for multiple events.
Step 4	**profile** *profile-name* **stop** **Example:** `RP/0/RSP0/CPU0:router (config-bs-schedule) # profile cpu-process stop`	Disables the profile.

Configuration Examples and Usecase Scenarios

The usecase scenarios with examples are discussed here.

Usecase-1: Collecting MIB Statistics

Goal: To collect IF MIB Statistics

Procedure	Example
Step1: Identifying the inputs and other parameters	MIB Objects of interest: • 1.3.6.1.2.1.2.2.1.2 (ifDescr) • 1.3.6.1.2.1.2.2.1.10 (ifInOctets) • 1.3.6.1.2.1.2.2.1.16 (ifOutOctets) Export Parameters: • Interval: 60 seconds • Protocol: TFTP • Server: 10.105.33.135 • Path: dcm_data

Step2: Configuring the **Data set if-mib**
For detailed procedure:
Configuring an SNMP Bulkstat Data Set, on page 321
	`bulkstat data if-mib type snmp`
	` object 1.3.6.1.2.1.2.2.1.2`
	` object 1.3.6.1.2.1.2.2.1.10`
	` object 1.3.6.1.2.1.2.2.1.16`

Procedure	Example
Step3: Configuring the **Instance set if-mib** For detailed procedure: Configuring an SNMP Bulkstat Instance Set, on page 323	`bulkstat instance if-mib type snmp` ` wildcard`

Step4: Configuring **Data Group if-group**
For detailed procedure:
Configuring a Bulkstat Data Group, on page 324
	`bulkstat data-group if-group`
	` interval polling 30`
	` collect type snmp data  if-mib instance`
	`if-mib`

Step5: Configuring **Profile snmp_profile**
For detailed procedure:
Configuring a Bulkstat Profile, on page 326
	`bulkstat profile snmp_profile`
	` file transfer url primary`
	`tftp://10.105.33.135/dcm_data/`
	` interval transfer raw 60`
	` data-group  if-group`
	` enable`

Note Step2 and Step3 can be interchanged.

Usecase-2: Using Filters

Goal: To collect gigabit ethernet interface statistics (using filters)

Procedure	Example
Step1: Identifying the inputs and other parameters	MIB Objects of interest: • 1.3.6.1.2.1.2.2.1.2 (ifDescr) • 1.3.6.1.2.1.2.2.1.10 (ifInOctets) • 1.3.6.1.2.1.2.2.1.16 (ifOutOctets) Export Parameters: • Interval: 60 seconds • Protocol: TFTP • Server: 10.105.33.135 • Path: dcm_data

Step2: Configuring the **Data set if-mib**
For detailed procedure:
Configuring an SNMP Bulkstat Data Set, on page 321
	`bulkstat data if-mib type snmp`
	` object 1.3.6.1.2.1.2.2.1.2`
	` object 1.3.6.1.2.1.2.2.1.10`
	` object 1.3.6.1.2.1.2.2.1.16`

Procedure	Example
Step3: Configuring the **Instance set if-mib** For detailed procedure: Configuring an SNMP Bulkstat Instance Set, on page 323	`bulkstat instance if-mib type snmp` ` wildcard`

Step4: Configuring the **Filter set if-mib**
For detailed procedure:
Configuring an SNMP Bulkstat Filter Set, on page 322
	Setting the filter (in this case, it is - gigabit ethernet interface)
	`bulkstat filter if-mib`
	`match 1.3.6.1.2.1.2.2.1.2 start`
	`"GigabitEthernet"`

Step5: Configuring **Data Group if-group**
For detailed procedure:
Configuring a Bulkstat Data Group, on page 324
	`bulkstat data-group if-group`
	` interval polling 30`
	` collect type snmp data  if-mib instance`
	`if-mib`

Step6: Configuring **Profile snmp_profile**
For detailed procedure:
Configuring a Bulkstat Profile, on page 326
	`bulkstat profile snmp_profile`
	` file transfer url primary`
	`tftp://10.105.33.135/dcm_data/`
	` interval transfer raw 60`
	` data-group  if-group`
	` enable`

Note Step2, Step3 and Step4 can interchanged.

Usecase-3: Collecting CLI output in XML format

Goal: To collect show cli output in XML format

Procedure	Example
Step1: Identifying the inputs and other parameters	CLI of interest: **add cmd show operational AAA xml** Export Parameters: • Interval: 5 minutes • Protocol: TFTP • Server: 10.64.68.12 • Path: dcm_data

Procedure	Example
Step2: Configuring the **Data set process** For detailed procedure: Configuring an SNMP Bulkstat Data Set, on page 321	`bulkstat data process type command` ` add cmd show operational AAA xml`

Step3: Configuring **Data Group cli-group**
For detailed procedure:
Configuring a Bulkstat Data Group, on page 324
	`bulkstat data-group  cli-group`
	` interval polling 60`
	` collect type command data sh snmp`

Step4: Configuring **Profile cli_profile**
For detailed procedure:
Configuring a Bulkstat Profile, on page 326
	`bulkstat profile cli_profile`
	` file transfer url primary`
	`tftp://10.64.68.12/dcm_data/`
	` interval transfer raw 300`
	` data-group cli-group`
	` enable`

CHAPTER 14

Configuring Frequency Synchronization

Frequency Synchronization is used to distribute precision frequency around a network. Frequency is synchronized accurately using Synchronized Ethernet (SyncE) in devices connected by Ethernet in a network.

This module describes the concepts around this and details the various configurations involved. For information on SyncE commands, see *System Management Command Reference for Cisco ASR 9000 Series Routers*.

This module contains the following topics:

Overview

Frequency or timing synchronization is the ability to distribute precision frequency around a network. In this context, timing refers to precision frequency, not an accurate time of day. Precision frequency is required in next generation networks for applications such as circuit emulation.

To achieve compliance to ITU specifications for TDM, differential method circuit emulation must be used, which requires a known, common precision frequency reference at each end of the emulated circuit. The incumbent example of frequency synchronization is provided by SDH equipment. This is used in conjunction with an external timing technology to provide synchronization of precision timing across the network.

SDH equipments are widely replaced by Ethernet equipments and synchronized frequency is required over such Ethernet ports. Synchronous Ethernet (SyncE) is used to accurately synchronize frequency in devices connected by Ethernet in a network. SyncE provides level frequency distribution of known common precision frequency references to a physical layer Ethernet network.

To maintain SyncE links, a set of operational messages are required. These messages ensure that a node is always deriving timing information from the most reliable source and then transfers the timing source quality information to clock the SyncE link. In SDH networks, these are known as Synchronization Status Messages (SSMs). SyncE uses Ethernet Synchronization Message Channel (ESMC) to provide transport for SSMs.

Source and Selection Points

Frequency Synchronization implementation involves Sources and Selection Points.

A Source inputs frequency signals into a system or transmits them out of a system. There are four types of sources:

- Line interfaces. This includes SyncE interfaces and SONET interfaces.
- Clock interfaces. These are external connectors for connecting other timing signals, such as BITS, UTI and GPS.
- PTP clock. If IEEE 1588 version 2 is configured on the router, a PTP clock may be available to frequency synchronization as a source of the time-of-day and frequency.
- Internal oscillator. This is a free-running internal oscillator chip.

Each source has a Quality Level (QL) associated with it which gives the accuracy of the clock. This QL information is transmitted across the network using ESMC or SSMs contained in the SDH frames. This provides information about the best available source the devices in the system can synchronize to. To define a predefined network synchronization flow and prevent timing loops, you can assign priority values to the sources on each router. The combination of QL information and user-assigned priority levels allow each router to choose a source to synchronize its SyncE or SDH interfaces, as described in the ITU standard G.781.

A Selection Point is any point where a choice is made between several frequency signals and possibly one or many of them are selected. Selection points form a graph representing the flow of timing signals between different cards in a router running Cisco IOS XR software. For example, there can be one or many selection points between different Synchronous Ethernet inputs available on a single line card. This information is forwarded to a selection point on the RSP, to choose between the selected source from each card.

The input signals to the selection points can be:

- Received directly from a source.
- Received as the output from another selection point on the same card.
- Received as the output from a selection point on a different card.

The output of a selection point can be used in a number of ways, like:

- To drive the signals sent out of a set of interfaces.
- As input into another selection point on a card.
- As input into a selection point on an another card.

Use **show frequency synchronization selection** command to see a detailed view of the different selection points within the system.

Note

- We recommend you to configure, and enable Frequency Synchronization selection input on two interfaces per line card.
- For link aggregation, you must configure and enable Frequency Synchronization selecction input on a single bundle member.

SyncE Hardware Support Matrix

This table provides details on the harware that supports SyncE:

Note The table also contains support details of upcoming releases. You can read this table in context of the current release and see relevant *Release Notes* for more information on supported features and hardware.

Table 44: Feature History Table

Feature Name	Release Information	Feature Description
SyncE Support on 5th Generation 10-Port 400 Gigabit Ethernet Line Cards: • A99-10X400GE-X-SE • A99-10X400GE-X-TR	Release 7.3.2	Frequency Synchronization is used to distribute precision frequency around a network. Frequency is synchronized accurately using Synchronized Ethernet (SyncE) in devices connected by Ethernet in a network. SyncE is now supported on the line cards: • A99-10X400GE-X-SE • A99-10X400GE-X-TR

Hardware Variant	Cisco IOS XR	Cisco IOS XR 64 bit
A9K-8X100GE-L-SE/TR (10GE and 100GE)	5.3.0	6.1.1
A9K-RSP880-SE/TR	5.3.0	6.1.1
A9K-8X100GE-L-SE/TR (40-GE)	6.0.1	6.1.1
A9K-4X100GE-SE/TR A9K-8X100GE-SE/TR	5.3.2 (100G LAN only) 6.0.1	6.1.1
A9K-MOD400-SE/TR A9K-MOD200-SE/TR with MPA 20x10GE and Legacy MPAs	6.0.1	6.2.2
A9K-MOD400-SE/TR A9K-MOD200-SE/TR with MPAs 2x100 and 1x100	6.1.3	6.2.2
A9K-400G-DWDM-TR	5.3.3 6.0.1	
A9K-24X10GE-1G-SE/TR A9K-48X10GE-1G-SE/TR	6.2.1	6.3.2

Hardware Variant	Cisco IOS XR	Cisco IOS XR 64 bit
A99-RSP-SE/TR (Cisco ASR 9910 Series Routers)	6.1.4	6.3.2
RSP880-LT-SE/TR	6.2.2	6.4.1
A9K-RSP440-TR/SE Enhanced Ethernet Linecards A99-RP-SE	4.3.4	
A99-RP2-TR/SE	5.3.0	6.3.2 6.4.1
Cisco ASR 9001 Series Routers	4.3.4	
Cisco ASR 9901 Series Routers	NA	6.4.1
A99-RSP-SE/TR (Cisco ASR 9906 Series Routers)	6.3.1	6.3.2
A9K-RSP5-SE/TR	NA	6.5.15
A99-RP3-SE/TR	NA	6.5.15
A9K-8X100GE-X-TR	NA	6.5.15
A9K-16X100GE-TR	NA	6.5.15
A9K-32X100GE-TR	NA	6.5.15
A99-32X100GE-X-TR	NA	7.1.15
A9K-8HG-FLEX-SE/TR	NA	7.1.15
A9K-20HG-FLEX-SE/TR	NA	7.1.15
ASR-9903	NA	7.1.3
A9903-20HG-PEC	NA	7.1.3
A99-10X400GE-X-SE/TR	NA	7.3.2
A99-12X100GE	NA	7.4.1
A9K-4X100GE	NA	7.4.1
ASR-9902	NA	7.4.1
A9K-4HG-FLEX-SE/TR	NA	7.4.1
A99-4HG-FLEX-SE/TR	NA	7.4.1

SyncE Restrictions

This section lists a few restrictions in configuring frequency synchronization. They are:

- On SyncE line interfaces, you can configure multiple interfaces for SyncE input. However, only one interface from each PHY gets selected as best source and programmed as SyncE input (there is no restriction on SyncE output) on the A9K-24X10GE-1G-SE/TR and A9K-48X10GE-1G-SE/TR line cards.
- *clock-interface timing-mode independent* configuration is not supported on Cisco A9K-RSP5-X-TR, A9K-RSP5-X-SE, A99-RP3-X-TR and A99-RP3-X-SE.

Clocking Support for nV Cluster

ASR9K cluster consists of two chassis connected together to provide redundancy and to meet higher bandwidth requirements. RSP440 provides two ICS (Inter-Chassis Synchronization) interfaces on the front plate. Clocking functionality support is added to the ICS interfaces. The ICS interfaces could be used for clocking, in the absence of other methods to synchronize frequency and Time-of-day information between the two cluster racks

nV Cluster Limitations

The limitations for the frequency synchronization support for cluster are:

- This feature is supported only on RSP440.
- The two chassis of the cluster have to be co-located. The length of the cable used for the ICS link should be less than 10 meters. This is needed to ensure the phase delay added due the length of the cable is within limits.
- SSM and QL is not supported on ICS links. SSM messages are not exchanged over the ICS interface. Hence, QL value needs to be configured under ICS clock interface configuration.
- The selection of an input clock source is based on the configuration of priority, QL as well as the clock quality. For SyncE, the ICS interfaces are similar to the SyncE line interfaces as far as input clock selection is concerned.
- All Input clock sources to cluster setup has to be redundant.
- No support for 1588 BC on LAG interfaces with member links across racks.

Inter-Chassis Synchronization (ICS)

ICS-Frequency Synchronization

Frequency synchronization is provided using Inter-Chassis Synchronization links (ICS). These are dedicated interfaces on the RSP used to synchronize the time and frequency.

The ICS link between the Primary DSC and Backup DSC carries the clock. There is no transfer of QL information from Primary DSC to Backup DSC.The clock direction is always from Primary DSC to Backup DSC. The Primary DSC transmits the clock and Backup DSC receives the clock.

The ICS clock interface (sync 2 or sync 3) is a clock input on the Backup DSC. The clock selection algorithm for SyncE is independent on each RSP. So, output clock from the rack which has Primary DSC is the outcome of the clock selection on the Primary DSC. The output clock from the rack which has Backup DSC is the outcome of the clock selection on the Backup DSC. If the ICS clock interface configuration is such that it is the selected clock on the Backup DSC, then the output clocks from the Primary rack and Backup rack are synchronised.

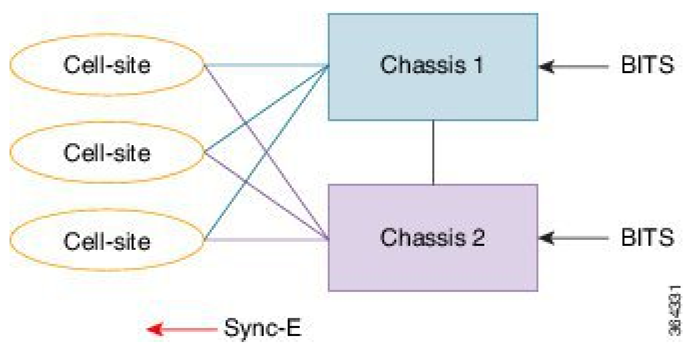
ICS-Time-of-Day

The ICS links also carry Time of Day (ToD) information when the ICS clock interfaces are configured for the same. Only the Backup DSC can synchronise with ToD from the Primary DSC and not vice versa. The 1588 clock information transmitted on all 1588 interfaces in the cluster (including interfaces on Backup rack) is of the clock selected at the Primary DSC. Thus, it is important that ICS clock interface on Backup DSC is configured such that it is the clock which is selected for ToD on the Backup DSC.

Recommended ICS Interface Connections

No inter-chassis frequency or time synchronization support:

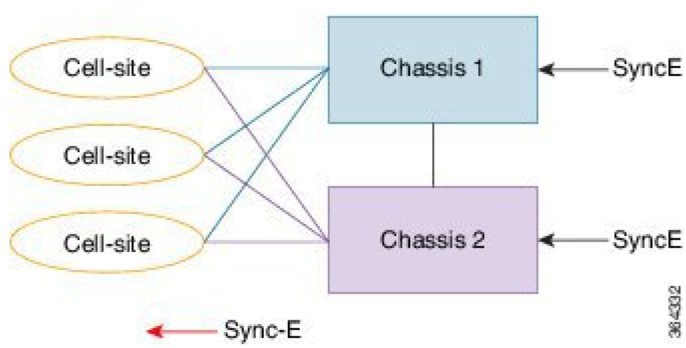
Figure 11: No inter-chassis frequency support



SyncE is used from the ASR9K cluster to provide precision frequency to mobile cell sites. A BITS clock is connected to each chassis of the cluster, meaning that the frequencies of both chassis are synchronized and the cell sites will all be synchronized, regardless of which chassis they synchronize to. In most deployments redundant BITS connections would be made to each chassis, to prevent against failure of any single BITS link.

With inter-chassis synchronization support:

Figure 12: With inter-chassis synchronization support



SyncE is used to synchronize the frequency of an ASR9k cluster to an upstream device. To provide redundancy in the case of one of the external SyncE inputs going down, the frequencies of the different cluster chassis must somehow be synchronized; else cell sites which select links from different chassis to synchronize may be out of sync if one of the SyncE links goes down.

Configuring Frequency Synchronization

Enabling Frequency Synchronization on the Router

This task describes the router-level configuration required to enable frequency synchronization.

Important **Frequency synchronization** configuration is mandatory to enable any type of frequency or time source on the router. This includes SyncE, clock-interfaces, Precision Time Protocol (PTP) or internal oscillator.

Note If timing mode system is not configured, the major alarm `T4 PLL is in FREERUN mode` is raised. This alarm has no functional impact to the system behavior.

SUMMARY STEPS

1. **configure**
2. **frequency synchronization**
3. **clock-interface timing-mode** {**independent** | **system**
4. **quality itu-t option** {**1** | **2 generation** {**1** | **2**}}
5. **log selection** {**changes** | **errors**}
6. Use one of these commands:
   - **end**
   - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**frequency synchronization** **Example:** `RP/0/RSP0/CPU0:router(config)# frequency synchronization`	Enables frequency synchronization on the router.

	Command or Action	Purpose
Step 3	**clock-interface timing-mode** {**independent** | **system** **Example:** `RP/0/RSP0/CPU0:router(config-freqsync)# clock-interface timing-mode system`	Configures the type of timing sources that can be used to drive the output from a clock interface. If this command is not used, the default quality mode is used. In the default mode, the clock interface output is driven only by input from line interfaces and the internal oscillator; it is never driven by input from another clock interface. In addition, some heuristic tests are run to detect if the signal being sent out of one clock interface can be looped back by some external box and sent back in via the same, or another clock interface. • **independent**—Specifies that the output of clock interfaces is driven only by the line interfaces (SyncE and SONET/SDH), as in the default mode. Loopback detection is disabled. • **system**—Specifies that the output of a clock interface is driven by the system-selected timing source (the source used to drive all SyncE and SONET/SDH interfaces), including clock interfaces. Loopback detection is disabled.
Step 4	**quality itu-t option** {**1** | **2 generation** {**1** | **2**}} **Example:** `RP/0/RSP0/CPU0:router(config-freqsync)# quality itu-t option 2 generation 1`	(Optional) Specifies the quality level for the router. The default is **option 1**. • **option 1**—Includes PRC, SSU-A, SSU-B, SEC and DNU. • **option 2 generation 1**—Includes PRS, STU, ST2, ST3, SMC, ST4, RES and DUS. • **option 2 generation 2**—Includes PRS, STU, ST2, ST3, TNC, ST3E, SMC, ST4, PROV and DUS. **Note** The quality option configured here must match the quality option specified in the **quality receive** and **quality transmit** commands in interface frequency synchronization configuration mode.
Step 5	**log selection** {**changes** | **errors**} **Example:** `RP/0/RSP0/CPU0:router(config-freqsync)# log selection changes`	Enables logging to frequency synchronization. • **changes**—Logs every time there is a change to the selected source, in addition to errors. • **errors**—Logs only when there are no available frequency sources, or when the only available frequency source is the internal oscillator.
Step 6	Use one of these commands: • **end** • **commit**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	Example:
	`RP/0/RSP0/CPU0:router(config-freqsync)# end`
	or
	`RP/0/RSP0/CPU0:router(config-freqsync)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **`yes`** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **`no`** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **`cancel`** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file, and remain within the configuration session.

What to do next

Configure frequency synchronization on any interfaces that should participate in frequency synchronization.

Configuring Frequency Synchronization on an Interface

By default, there is no frequency synchronization on line interfaces. Use this task to configure an interface to participate in frequency synchronization.

Before you begin

You must enable frequency synchronization globally on the router.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **frequency synchronization**
4. **selection input**
5. **priority** *priority-value*
6. **wait-to-restore** *minutes*
7. **ssm disable**
8. **time-of-day-priority** *priority*
9. **quality transmit** {**exact** | **highest** | **lowest**} **itu-t option** *ql-option*
10. **quality receive** {**exact** | **highest** | **lowest**} **itu-t option** *ql-option*
11. Use one of these commands:
    - **end**
    - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/1/1/0`	Enters interface configuration mode.
Step 3	**frequency synchronization** **Example:** `RP/0/RSP0/CPU0:router(config-if)# frequency synchronization`	Enables frequency synchronization on the interface and enters interface frequency synchronization mode to configure the various options. By default, this causes the system selected frequency signal to be used for clocking transmission, but does not enable the use of the interface as an input.
Step 4	**selection input** **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# selection input`	(Optional) Specifies the interface as a timing source to be passed to the selection algorithm.
Step 5	**priority** *priority-value* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# priority 100`	(Optional) Configures the priority of the frequency source on a controller or an interface. Values can range from 1 (highest priority) to 254 (lowest priority). The default value is 100. This command is used to set the priority for an interface or clock interface. The priority is used in the clock-selection algorithm to choose between two sources that have the same quality level (QL). Lower priority values are preferred.
Step 6	**wait-to-restore** *minutes* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# wait-to-restore 300`	(Optional) Configures the wait-to-restore time, in minutes, for frequency synchronization on an interface. This is the amount of time after the interface comes up before it is used for synchronization. Values can range from 0 to 12. The default value is 5.
Step 7	**ssm disable** **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# ssm disable`	(Optional) Disables Synchronization Status Messages (SSMs) on the interface. • For SyncE interfaces, this disables sending ESMC packets, and ignores any received ESMC packets. • For SONET and clock interfaces, this causes DNUs to be sent, and ignores any received QL value.

	Command or Action	Purpose
Step 8	**time-of-day-priority** *priority* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# time-of-day-priority 50`	(Optional) Specifies the priority of this time source as the time-of-day (ToD) source. The priority is used as the first criterion when selecting between sources for a time-of-day selection point. Values can range from 1 (highest priority) to 254 (lowest priority); the default value is 100.
Step 9	**quality transmit** {**exact** | **highest** | **lowest**} **itu-t option** *ql-option* **Example:** `RP/0/RSP0/CPU0:router(config-clk-freqsync)# quality transmit highest itu-t option 1 prc`	(Optional) Adjusts the QL that is transmitted in SSMs. • **exact** *ql*—Specifies the exact QL to send, unless DNU would otherwise be sent. • **highest** *ql*—Specifies an upper limit on the QL to be sent. If the selected source has a higher QL than the QL specified here, this QL is sent instead. • **lowest** *ql*—Specifies a lower limit on the QL to be sent. If the selected source has a lower QL than the QL specified here, DNU is sent instead. The quality option specified in this command must match the globally-configured quality option in the **quality itu-t option** command. **Note** For clock interfaces that do not support SSM, only the lowest QL can be specified. In this case, rather than sending DNU, the output is squelched, and no signal is sent.
Step 10	**quality receive** {**exact** | **highest** | **lowest**} **itu-t option** *ql-option* **Example:** `RP/0/RSP0/CPU0:router(config-clk-freqsync)# quality receive highest itu-t option 1 prc`	(Optional) Adjusts the QL value that is received in SSMs, before it is used in the selection algorithm. • **exact** *ql*—Specifies the exact QL regardless of the value received, unless the received value is DNU. • **highest** *ql*—Specifies an upper limit on the received QL. If the received value is higher than this specified QL, this QL is used instead. • **lowest** *ql*—Specifies a lower limit on the received QL. If the received value is lower than this specified QL, DNU is used instead. The quality option specified in this command must match the globally-configured quality option in the **quality itu-t option** command. **Note** For clock interfaces that do not support SSM, only the exact QL can be specified.
Step 11	Use one of these commands: • **end** • **commit**	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes:

	Command or Action	Purpose

	Example:
	`RP/0/RSP0/CPU0:router(config-if-freqsync)# end`
	or
	`RP/0/RSP0/CPU0:router(config-if-freqsync)# commit`
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **`yes`** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **`no`** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **`cancel`** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file, and remain within the configuration session.

Configuring Frequency Synchronization on a Clock Interface

To enable a clock interface to be used as frequency input or output, you must configure the port parameters and frequency synchronization, as described in this task.

> **Note** The configuration on clock interfaces must be the same for corresponding clock interfaces across all RSPs to avoid changes in frequency synchronization behavior in the event of an RSP switchover.

SUMMARY STEPS

1. **configure**
2. **clock-interface sync** *port-no* **location** *node-id*
3. **port-parameters** {**bits-input** *mode* | **bits-output** *mode* | **dti**}
4. **ics**
5. **frequency synchronization**
6. **selection input**
7. **priority** *priority-value*
8. **wait-to-restore** *minutes*
9. **ssm disable**
10. **time-of-day-priority** *priority*
11. **quality transmit** {**exact** | **highest** | **lowest**} **itu-t option** *ql-option*
12. **quality receive** {**exact** | **highest** | **lowest**} **itu-t option** *ql-option*
13. Use one of these commands:
    - **end**

- **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**clock-interface sync** *port-no* **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router(config)# clock-interface sync 2 location 0/2/0`	Enters clock interface configuration mode to configure the clock interface.
Step 3	**port-parameters** {**bits-input** *mode* | **bits-output** *mode* | **dti**} **Example:** `RP/0/RSP0/CPU0:router(config-clock-if)# port-parameters dti`	Specifies the type of external clock source for the clock interface. Options are BITS RX, BITS TX or DTI. The possible *mode* values for BITS interfaces are **2m**, **6m-output-only**, **e1** or **t1**.
Step 4	**ics** **Example:** `RP/0/RSP0/CPU0:router(config)# ics`	Enables chassis synchronization.

Step 5	**frequency synchronization**
	Example:
	`RP/0/RSP0/CPU0:router(config-clock-if)# frequency synchronization`
	`RP/0/RSP0/CPU0:router(config-clk-freqsync)#`
		Enters clock interface frequency synchronization mode to configure frequency synchronization parameters.
		Note The remaining steps in this task are the same as those used to configure the interface frequency synchronization.

Step 6	**selection input** **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# selection input`	(Optional) Specifies the interface as a timing source to be passed to the selection algorithm.
Step 7	**priority** *priority-value* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# priority 100`	(Optional) Configures the priority of the frequency source on a controller or an interface. Values can range from 1 (highest priority) to 254 (lowest priority). The default value is 100. This command is used to set the priority for an interface or clock interface. The priority is used in the clock-selection algorithm to choose between two sources that have the same quality level (QL). Lower priority values are preferred.

	Command or Action	Purpose
Step 8	**wait-to-restore** *minutes* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# wait-to-restore 300`	(Optional) Configures the wait-to-restore time, in minutes, for frequency synchronization on an interface. This is the amount of time after the interface comes up before it is used for synchronization. Values can range from 0 to 12. The default value is 5.
Step 9	**ssm disable** **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# ssm disable`	(Optional) Disables Synchronization Status Messages (SSMs) on the interface. • For SyncE interfaces, this disables sending ESMC packets, and ignores any received ESMC packets. • For SONET and clock interfaces, this causes DNUs to be sent, and ignores any received QL value.
Step 10	**time-of-day-priority** *priority* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# time-of-day-priority 50`	(Optional) Specifies the priority of this time source as the time-of-day (ToD) source. The priority is used as the first criterion when selecting between sources for a time-of-day selection point. Values can range from 1 (highest priority) to 254 (lowest priority); the default value is 100.
Step 11	**quality transmit** {**exact** | **highest** | **lowest**} **itu-t option** *ql-option* **Example:** `RP/0/RSP0/CPU0:router(config-clk-freqsync)# quality transmit highest itu-t option 1 prc`	(Optional) Adjusts the QL that is transmitted in SSMs. • **exact** *ql*—Specifies the exact QL to send, unless DNU would otherwise be sent. • **highest** *ql*—Specifies an upper limit on the QL to be sent. If the selected source has a higher QL than the QL specified here, this QL is sent instead. • **lowest** *ql*—Specifies a lower limit on the QL to be sent. If the selected source has a lower QL than the QL specified here, DNU is sent instead. The quality option specified in this command must match the globally-configured quality option in the **quality itu-t option** command. **Note** For clock interfaces that do not support SSM, only the lowest QL can be specified. In this case, rather than sending DNU, the output is squelched, and no signal is sent.
Step 12	**quality receive** {**exact** | **highest** | **lowest**} **itu-t option** *ql-option* **Example:** `RP/0/RSP0/CPU0:router(config-clk-freqsync)# quality receive highest itu-t option 1 prc`	(Optional) Adjusts the QL value that is received in SSMs, before it is used in the selection algorithm. • **exact** *ql*—Specifies the exact QL regardless of the value received, unless the received value is DNU. • **highest** *ql*—Specifies an upper limit on the received QL. If the received value is higher than this specified QL, this QL is used instead.

	Command or Action	Purpose
		• **lowest** *ql*—Specifies a lower limit on the received QL. If the received value is lower than this specified QL, DNU is used instead. The quality option specified in this command must match the globally-configured quality option in the **quality itu-t option** command. **Note** For clock interfaces that do not support SSM, only the exact QL can be specified.

Step 13	Use one of these commands:
	• **end**
	• **commit**
	
	Example:
	
	`RP/0/RSP0/CPU0:router(config-if-freqsync)# end`
	
	or
	
	`RP/0/RSP0/CPU0:router(config-if-freqsync)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		
		  • Entering **`yes`** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		  • Entering **`no`** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		  • Entering **`cancel`** leaves the router in the current configuration session without exiting or committing the configuration changes.
		
		• Use the **commit** command to save the configuration changes to the running configuration file, and remain within the configuration session.

Configuring Clock Interface with DTI input

This procedure describes the steps involved to configure a Clock interface with DTI input.

1. To configure a clock interface, use **clock-interface sync** *value* **location** *node* command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# clock-interface sync 1 location 0/RSP0/CPU0
```

2. To configure port parameters for the given clock interface, use **port-parameters dti** command in the clock-interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-clock-if)# port-parameters dti
```

3. To enable frequency synchronization, use **frequency synchronization** command in the clock-interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-clock-if)# frequency synchronization
```

4. To configure selection input for the given clock interface, use **selection input** command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# selection input
```

5. To configure priority for the clock interface, use **priority** *number* command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# priority 1
```

6. To configure wait-to-restore time for the clock interface, use **wait-to-restore** *number* command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# wait-to-restore 0
```

7. To disable SSM packets for the clock interface, use **ssm disable** command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# ssm disable
```

8. To configure quality settings for the clock interface, use **quality receive exact itu-t option** *number* **generation** *number* **PRS** command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# quality receive exact itu-t option 2
generation 2 PRS
```

Verification

To display the current running configuration of an interface, use **show run clock-interface** command.

```
RP/0/RSP0/CPU0:router# show run clock-interface sync 1 location 0/RSP0/CPU0

clock-interface sync 1 location 0/RSP0/CPU0
port-parameters
  dti
!
frequency synchronization
  selection input
  priority 1
  wait-to-restore 0
  ssm disable
  quality receive exact itu-t option 2 generation 2 PRC
!
!RP/0/RSP0/CPU0:router#
```

Configuring GPS Settings for a sync2 interface

This procedure describes the steps involved to configure GPS settings for a sync2 interface.

1. To configure a clock interface, use **clock-interface sync** *port-number* **location** *interface-location* command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# clock-interface sync 2 location 0/RSP0/CPU0
```

2. To configure port parameters for the given clock interface, use **port-parameters** command in the clock-interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-clock-if)# port-parameters
```

3. To configure GPS input parameters, use **gps-input tod-format gprmc pps-input ttl** command.

```
RP/0/RSP0/CPU0:router(config-clk-parms)# gps-input tod-format
gprmc pps-input ttl
```

4. To return to the clock-interface configuration mode, use **exit** command.

```
RP/0/RSP0/CPU0:router(config-clk-parms)# exit
```

5. To enable frequency synchronization, use **frequency synchronization** command in the clock-interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-clock-if)# frequency synchronization
```

6. To configure selection input for the given clock interface, use **selection input** command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# selection input
```

7. To configure priority for the clock interface, use **priority** *number* command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# priority 10
```

8. To configure wait-to-restore time for the clock interface, use **wait-to-restore** *number* command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# wait-to-restore 0
```

9. To disable SSM packets for the clock interface, use **ssm disable** command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# ssm disable
```

10. To configure quality settings for the clock interface, use **quality receive exact itu-t option** *number* **generation** *number* **PRS** command in the frequency-synchronization clock-configuration mode.

```
RP/0/RSP0/CPU0:router(config-clk-freqsync)# quality receive exact itu-t option 2
generation 2 PRS
```

Verification

To verify the configured GPS parameters , use **show run clock-interface** command.

```
RP/0/RSP0/CPU0:router# show run clock-interface sync 2 location 0/RSP0/CPU0

clock-interface sync 2 location 0/RSP0/CPU0
port-parameters
gps-input tod-format gprmc pps-input ttl
!
```

GPS ToD Support for NMEA

National Marine Electronics Associations (NMEA) 0183 is a standard protocol used by GPS receivers to transmit data and is responsible for creating a standard uniform interface for digital data exchange between different marine electronic products. NMEA provides protocol strings to send out GPS updates. GPRMC is one such NMEA string that provides exact data and time (Greenwich time), latitude, longitude, heading, and speed. Router receives GPS ToD messages in serial ASCII stream through the RS422 interface in three formats - NTP Type 4, Cisco, and GPRMC. The timing data is extracted from this stream.

Note Cisco ASR 9000 Series Routers can support ToD in NMEA or GPRMC format. Currently, this is supported only on RS422.

Note You can refer to the below support information in context of the current release and see relevant *Release Notes* for more information on supported features and hardware.

Supported hardware are:

- A9K-RSP440-SE/TR
- A9K-RSP880-SE/TR
- A99-RP2-SE/TR
- A9K-RSP880-LT-SE/TR
- A99-RSP-SE/TR

Configuring ICS

This task enables inter-chassis synchronization for interfaces.

SUMMARY STEPS

1. **configure**
2. **clock-interface sync** *port-no* **location** *node-id*
3. **port-parameters ics**
4. **frequency synchronization**
5. **selection input**
6. **priority** *priority-value*
7. **wait-to-restore** *minutes*

8. **time-of-day-priority** *priority*
9. **quality receive { exact | highest | lowest} itu-t option** *option*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**clock-interface sync** *port-no* **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router(config)# clock-interface sync 2 location 1/RSP0/CPU0`	Enters clock interface configuration mode to configure the clock interface.
Step 3	**port-parameters ics** **Example:** `RP/0/RSP0/CPU0:router(config-clock-if)# port-parameters ics`	Enables inter-chassis synchronization.

Step 4	**frequency synchronization**
	Example:
	`RP/0/RSP0/CPU0:router(config-clock-if)# frequency synchronization`
	`RP/0/RSP0/CPU0:router(config-clk-freqsync)#`
		Enters clock interface frequency synchronization mode to configure frequency synchronization parameters.
		Note The remaining steps in this task are the same as those used to configure the interface frequency synchronization.

Step 5	**selection input** **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# selection input`	(Optional) Specifies the interface as a timing source to be passed to the selection algorithm.
Step 6	**priority** *priority-value* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# priority 100`	(Optional) Configures the priority of the frequency source on a controller or an interface. Values can range from 1 (highest priority) to 254 (lowest priority). The default value is 100. This command is used to set the priority for an interface or clock interface. The priority is used in the clock-selection algorithm to choose between two sources that have the same quality level (QL). Lower priority values are preferred.
Step 7	**wait-to-restore** *minutes* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# wait-to-restore 300`	(Optional) Configures the wait-to-restore time, in minutes, for frequency synchronization on an interface. This is the amount of time after the interface comes up before it is used for synchronization. Values can range from 0 to 12. The default value is 5.

	Command or Action	Purpose
Step 8	**time-of-day-priority** *priority* **Example:** `RP/0/RSP0/CPU0:router(config-if-freqsync)# time-of-day-priority 50`	(Optional) Specifies the priority of this time source as the time-of-day (ToD) source. The priority is used as the first criterion when selecting between sources for a time-of-day selection point. Values can range from 1 (highest priority) to 254 (lowest priority); the default value is 100.
Step 9	**quality receive { exact | highest | lowest} itu-t option** *option* **Example:** `RP/0/RSP0/CPU0:router (config-clk-freqsync) #` **`quality receive exact itu-t option 1 PRC`**	

Verifying the Frequency Synchronization Configuration

After performing the frequency synchronization configuration tasks, use this task to check for configuration errors and verify the configuration.

SUMMARY STEPS

1. **show frequency synchronization configuration-errors**
2. **show frequency synchronization interfaces brief**
3. **show frequency synchronization interfaces** *node-id*
4. **show processes fsyncmgr location** *node-id*

DETAILED STEPS

Step 1 **show frequency synchronization configuration-errors**

Example:

```
RP/0/RSP0/CPU0:router# show frequency synchronization configuration-errors

Node 0/2/CPU0:
==============
  interface GigabitEthernet0/2/0/0 frequency synchronization
    * Frequency synchronization is enabled on this interface, but isn't enabled globally.

  interface GigabitEthernet0/2/0/0 frequency synchronization quality transmit exact itu-t option 2
generation 1 PRS
    * The QL that is configured is from a different QL option set than is configured globally.
```

Displays any errors that are caused by inconsistencies between shared-plane (global) and local-plane (interface) configurations. There are two possible errors that can be displayed:

- Frequency Synchronization is configured on an interface (line interface or clock-interface), but is not configured globally. Refer to Enabling Frequency Synchronization on the Router, on page 339
- The QL option configured on some interface does not match the global QL option. Under an interface (line interface or clock interface), the QL option is specified using the **quality transmit** and **quality receive** commands. The value

specified must match the value configured in the global **quality itu-t option** command, or match the default (option 1) if the global **quality itu-t option** command is not configured.

Once all the errors have been resolved, meaning there is no output from the command, continue to the next step.

Step 2 **show frequency synchronization interfaces brief**

Example:

```
RP/0/RSP0/CPU0:router# show frequency synchronization interfaces brief

Flags:  > - Up                D - Down              S - Assigned for selection
        d - SSM Disabled      x - Peer timed out    i - Init state

Fl  Interface                QLrcv QLuse Pri QLsnt Source
=== ======================== ===== ===== === ===== ========================
>Sx GigabitEthernet0/2/0/0   Fail  Fail  100 DNU   None
Dd  GigabitEthernet0/2/0/1   n/a   Fail  100 n/a   None

RP/0/RSP0/CPU0:router# show frequency synchronization clock-interfaces brief

Flags:  > - Up                D - Down              S - Assigned for selection
        d - SSM Disabled      s - Output squelched  L - Looped back

Node 0/0/CPU0:
==============
  Fl    Clock Interface     QLrcv  QLuse  Pri QLsnd  Source
  ===== =================== ====== ====== === ====== ========================
  >S    Sync0               PRC    Fail   100 SSU-B  Internal0 [0/0/CPU0]
  >     Sync1               SSU-A  Fail   100 SSU-B  Internal0 [0/0/CPU0]
  >S    Internal0           n/a    SSU-B  255 n/a    None

Node 0/1/CPU0:
==============
  Fl    Clock Interface     QLrcv  QLuse  Pri QLsnd  Source
  ===== =================== ====== ====== === ====== ========================
  D     Sync0               None   Fail   100 SSU-B  Internal0 [0/1/CPU0]
  D     Sync1               None   Fail   100 SSU-B  Internal0 [0/1/CPU0]
  >S    Internal0           n/a    SSU-B  255 n/a    None
```

Verifies the configuration. Note the following points:

- All line interface that have frequency synchronization configured are displayed.
- All clock interfaces and internal oscillators are displayed.
- Sources that have been nominated as inputs (in other words, have **selection input** configured) have ‘S’ in the Flags column; sources that have not been nominated as inputs do not have ‘S’ displayed.

  Note Internal oscillators are always eligible as inputs.

- ‘>’ or ‘D’ is displayed in the flags field as appropriate.

If any of these items are not true, continue to the next step.

Step 3 **show frequency synchronization interfaces** *node-id*

Example:

```
RP/0/RSP0/CPU0:router# show frequency synchronization interfaces GigabitEthernet0/2/0/2
```

```
Interface GigabitEthernet0/2/0/2 (shutdown)
  Assigned as input for selection
  SSM Enabled
  Input:
    Down
    Last received QL: Failed
    Effective QL:     Failed, Priority: 100
  Output:
    Selected source:    Sync0 [0/0/CPU0]
    Selected source QL: Opt-I/PRC
    Effective QL:       Opt-I/PRC
  Next selection points: LC_INGRESS

RP/0/RSP0/CPU0:router# show frequency synchronization clock-interfaces location 0/1/CPU0

Node 0/1/CPU0:
==============
  Clock interface Sync0 (Down: mode not configured)
    SSM supported and enabled
    Input:
      Down
      Last received QL: Opt-I/PRC
      Effective QL:     Failed, Priority: 100
    Output:
      Selected source:    Internal0 [0/1/CPU0]
      Selected source QL: Opt-I/SSU-B
      Effective QL:       Opt-I/SSU-B
  Next selection points: RP_SYSTEM

  Clock interface Sync1 (Down: mode not configured)
    SSM supported and enabled
    Input:
      Down
      Last received QL: Opt-I/PRC
      Effective QL:     Failed, Priority: 100
    Output:
      Selected source:    Internal0 [0/1/CPU0]
      Selected source QL: Opt-I/SSU-B
      Effective QL:       Opt-I/SSU-B
  Next selection points: RP_SYSTEM

  Clock interface Internal0 (Up)
    Assigned as input for selection
    Input:
      Default QL:   Opt-I/SSU-B
      Effective QL: Opt-I/SSU-B, Priority: 255
  Next selection points: RP_SYSTEM RP_CLOCK_INTF
```

Investigates issues within individual interfaces. If the clock interface is down, a reason is displayed. This may be because there is missing or conflicting platform configuration on the clock interface.

Step 4 **show processes fsyncmgr location** *node-id*

Example:

```
RP/0/RSP0/CPU0:router# show processes fsyncmgr location 0/0/CPU0

                  Job Id: 134
                     PID: 30202
         Executable path: /pkg/bin/fsyncmgr
              Instance #: 1
```

```
             Version ID: 00.00.0000
                Respawn: ON
          Respawn count: 1
  Max. spawns per minute: 12
           Last started: Mon Mar  9 16:30:43 2009
          Process state: Run
          Package state: Normal
      Started on config: cfg/gl/freqsync/g/a/enable
                   core: MAINMEM
              Max. core: 0
              Placement: None
           startup_path: /pkg/startup/fsyncmgr.startup
                  Ready: 0.133s
       Process cpu time: 1730768.741 user, -133848.-361 kernel, 1596920.380 total
  --------------------------------------------------------------------------------
```

Verifies that the fsyncmgr process is running on the appropriate nodes.

Class C Timing Support

To Configure Class C mode, see Class C Timing Mode.

CHAPTER 15

Configuring Precision Time Protocol

Precision Time Protocol (PTP) is a protocol that defines a method to distribute time around a network. PTP support is based on the IEEE 1588-2008 standard.

This module describes the concepts around this protocol and details the various configurations involved. For information on PTP commands, see *System Management Command Reference for Cisco ASR 9000 Series Routers*.

This module contains the following topics:

Overview

The Precision Time Protocol (PTP), as defined in the IEEE 1588 standard, synchronizes with nanosecond accuracy the real-time clocks of the devices in a network. The clocks are organized into a server-client hierarchy. PTP identifies the port that is connected to a device with the most precise clock. This clock is referred to as the server clock. All the other devices on the network synchronize their clocks with the server and are referred to as members. Constantly-exchanged timing messages ensure continued synchronization. PTP ensures that the best available clock is selected as the source of time (the grandmaster clock) for the network and that other clocks in the network are synchronized to the grandmaster.

Table 45: PTP Clocks

Network Element	Description
Grandmaster (GM)	A network device physically attached to the primary time source. All clocks are synchronized to the grandmaster clock.

Network Element	Description
Ordinary Clock (OC)	An ordinary clock is a 1588 clock with a single PTP port that can operate in one of the following modes: • server mode—Distributes timing information over the network to one or more client clocks, thus allowing the client to synchronize its clock to the server. • client mode—Synchronizes its clock to a server clock. You can enable the client mode on up to two interfaces simultaneously in order to connect to two different server clocks.
Boundary Clock (BC)	The device participates in selecting the best server clock and can act as the server clock if no better clocks are detected. Boundary clock starts its own PTP session with a number of downstream clients. The boundary clock mitigates the number of network hops and results in packet delay variations in the packet network between the Grandmaster and client.
Transparent Clock (TC)	A transparent clock is a device or a switch that calculates the time it requires to forward traffic and updates the PTP time correction field to account for the delay, making the device transparent in terms of time calculations.

PTP consists of two parts:

- The port State machine and Best Master Clock Algorithm: This provides a method to determine the ports in the network that will remain passive (neither server nor client), run as a server (providing time to other clocks in the network), or run as clients (receiving time from other clocks in the network).
- Delay-Request/Response mechanism and a Peer-delay mechanism: This provides a mechanisms for client ports to calculate the difference between the time of their own clocks and the time of their server clock.

Note Cisco ASR 9000 Series routers do not support Peer-delay mechanism.

The implementation of PTP on Cisco IOS XR software is designed to operate effectively in Telecommunication networks, which are different from the networks for which PTP was originally designed.

PTP is configurable on Gigabit Ethernet interfaces (1G, 10G, 40G, and 100G), Bundle Ethernet interfaces, and sub-interfaces. PTP is not configurable on LAG Ethernet sub-interfaces.

Frequency and Time Selection

The selection of the source to synchronize the backplane clock frequency is made by frequency synchronization, and is outside of the scope of PTP. The Announce, Sync, and Delay-request frequencies must be the same on the server and client.

Delay-Response Mechanism

The Delay Request-response mechanism (defined in section 11.3 of IEEE Std 1588-2008) lets a client port estimate the difference between its own clock-time and the clock-time of its server. The following options are supported:

- One-step mechanism - The timestamp for a Sync message is sent in the Sync message itself.
- Two-step mechanism - The timestamp for a Sync message is sent later in a Follow-up message.

When running a port in client state, a router can send Delay-request messages and handle incoming Sync, Follow-up, and Delay-response messages. The timeout periods for both Sync and Delay-response messages are individually configurable.

Hybrid Mode

Your router allows the ability to select separate sources for frequency and time-of-day (ToD). Frequency selection can be between any source of frequency available to the router, such as: BITS, GPS, SyncE or IEEE 1588 PTP. The ToD selection is between the source selected for frequency and PTP, if available (ToD selection is from GPS, DTI or PTP). This is known as hybrid mode, where a physical frequency source (BITS or SyncE) is used to provide frequency synchronization, while PTP is used to provide ToD synchronization.

Frequency selection uses the algorithm described in ITU-T recommendation G.871, and is described in the *Configuring Frequency Synchronization* module in this document. The ToD selection is controlled using the time-of-day priority configuration. This configuration is found under the source interface frequency synchronization configuration mode and under the global PTP configuration mode. It controls the order for which sources are selected for ToD. Values in the range of 1 to 254 are allowed, with lower numbers indicating higher priority.

Port States

State machine indicates the behavior of each port. The possible states are:

State	Description
INIT	Port is not ready to participate in PTP.
LISTENING	First state when a port becomes ready to participate in PTP: In this state, the port listens to PTP servers for a (configurable) period of time.
PRE-MASTER	Port is ready to enter the Server state.
MASTER	Port provides timestamps for any client or boundary clocks that are listening.
UNCALIBRATED	Port receives timestamps from a server clock but, the router's clock is not yet synchronized to the server.

State	Description
SLAVE	Port receives timestamps from a server clock and the router's clock is synchronized to the server.
PASSIVE	Port is aware of a better clock than the one it would advertise if it was in server state and is not a client clock to that server clock.

Leap Seconds

In prior releases, IOS-XR only offered a static and time-consuming solution to manage leap seconds. For every upcoming leap second inclusion, the number of leap seconds had to be hard-coded into a Software Maintenance Update (SMU) and also installed on the router for the same. It is a prolonged and tedious process to provide and install a SMU each time a new leap second is announced.

From Release 6.4.1 onward, Cisco IOS-XR supports leap-second configuration instead of SMU installations or reloads.

Time is measured using a common timescale. Leap second factor is used to adjust the current time to compensate for any drift from the common timescale. Leap seconds are introduced to dynamically adjust the UTC offset in response to leap second events. The two most relevant timescales are:

- **TAI - International Atomic Time** : This is a notional passage of time determined by weighted average of readings across a large number of atomic clocks.
- **UTC - Universal Coordinated Time** : This differs from TAI by an integer number of seconds to remain in synchronization with mean solar time. UTC is related to a notion of time called **UT1**, which represents the mean solar time at 0° longitude. Leap seconds are periodically inserted to ensure UTC and UT1 are never more than 0.9 seconds apart.

PTP uses TAI timescale. UTC time is derived using UTC offset. UTC offset and the number of seconds in the last minute of the current UTC day are sent in the PTP header of Announce messages.

UTC is calculated as: **UTC = TAI - offset**.

IOS-XR PTP implementation uses the following sources (in order of decreasing precedence) to determine the current UTC offset value:

- The current grandmaster clock, if present.
- UTC offset configuration, if present.
- The previous grandmaster clock, if one exists.
- The hardware (e.g. a locally connected GPS receiver), if available.
- Zero, indicating that no UTC offset information is available.

If any upcoming leap second (being advertised at the time synchronization with a grandmaster) is lost, that too will be applied at the appropriate time while in holdover

Note

- Leap seconds are generally added by including an extra second (23:59:60), either on June 30th or on December 31st.
- UTC offset is + 37 seconds, as of 01 Jan 2017.

Multiple PTP Profile Interoperability

Communication between two different profiles was not possible previously due to various factors like, incompatible domain numbers, BMCA, or clock-class leading to drop in packets. Also, you cannot compare devices running different profiles in such configurations. For example, the domain number for G.8275.1 profile (24) is incompatible with the domain number for G.8275.2 profile (44).

Multiple PTP Profile Interoperability feature lets you develop a configuration to communicate with a peer device running a different PTP profile than the profile that is configured on the source router. This means that multiple profiles can interoperate on a single device in this implementation.

Interoperation is achieved by converting packets on ingress/egress so that it is acceptable to the profile configured on the receiving device. This prevents packet loss and allows comparison of different profiles. You can configure the interoperation using the **interop** command. Configuration details are described in a later section in this chapter. For command details, refer to Precision Time Protocol (PTP) Commands chapter in the *System Management Command Reference for Cisco ASR 9000 Series Routers* guide.

Note

- Multiple ingress conversions are performed for interfaces configured with multiple servers.
- Only G.8275.1 and G.8275.2 profiles can be configured to interoperate.

Class C Timing Mode

Table 46: Feature History Table

Feature Name	Release Information	Feature Description
Class C Timing Mode	Release 7.6.2	We now support the enhanced timing mode, Class C, as per the revised version of G.8273.2 by ITU-T. You can now switch to this Class C mode to avail high-accuracy clocks in the telecom networks having precise timing requirements, such as 5G networks. Class C mode reduces the Maximum Absolute Time Error (Max|TE|) and enhances synchronization of Telecom Boundary Clock (T-BC) and Telecom Time Secondary Clock (T-TSC). Class C timing support is available for both PTP and Frequency Synchronization. Class C timing mode is supported only on the routers with the combination of following Route Switch Processors (RSPs) and Fifth generation of the ASR 9000 Series Ethernet line cards except A99-32X100GE-X-TR/SE: • A9K-RSP5-X-TR/SE • A99-RP3-X-TR/SE This feature introduces the timing-accuracy enhanced command.

The advent of 5G technology demands strict timing requirements. To satisfy the strict timing requirements, ITU-T has introduced Class C or Enhanced timing accuracy mode under G8273.2 profile. Class B or Legacy mode is the default mode. The Max|TE| for Class B is 70 ns, whereas for Class C it is 40 ns. Reduced Max|TE| error indicates reduced noise transfer and improved timing accuracy between T-BC and T-SC.

Restrictions

- Class C timing support is not available on A99-32X100GE-X-TR/SE Lightspeed-plus-based line card.
- Class C is not supported on the following line cards:
  - Third generation of the ASR 9000 Series Ethernet Line Cards

- BITS-output clock does not work with Cisco IOS XR Software Release 7.6.2 image. You must install Software Maintenance Updates (SMU) for the same.
- If the **timing-accuracy enhanced** command is applied on the router with unsupported line cards, the timing functionality stops working on those line cards.

  The following system log appears on the router console when Class C configuration is applied on the routers containing unsupported hardware that does not support Class C mode.

```
LC/0/1/CPU0:Sep 2 15:51:50.791 UTC: sync_agent[325]:
%PLATFORM-FSYNC-4-CLOCK_ACCURACY_UNSUPPORTED : This line card does not support enhanced
 accuracy mode
```

To configure Class C mode, see Configuring Class C Timing Mode, on page 399.

PTP Phase Difference Threshold Between Passive and Secondary Ports

Table 47: Feature History Table

Feature Name	Release Information	Feature Description
PTP Phase Difference Threshold Between Passive and Secondary Ports	Release 24.2.1	Passive ports can now be included in the Delay Request-Response Mechanism (DRRM), which allows for the monitoring of PTP phase differences between a passive port and a secondary port. If these PTP phase differences surpass a predefined limit, system logs are triggered. This feature enables you to detect potential errors such as fiber asymmetry or a clock failure in the PTP network. This feature introduces these changes: **CLI:** • **phase-difference-threshold-breach** • The **show ptp foreign-masters** command output is enhanced to include phase difference values and servo status. **YANG Data Models:** The following data models are enhanced: • `Cisco-IOS-XR-ptp-cfg.yang` • `Cisco-IOS-XR-um-ptp-cfg.yang`

The Precision Time Protocol (PTP), as defined in the IEEE 1588 standard, is designed for precise time synchronization across networked devices. It operates by having Foreign Masters (FMs) broadcast timing information to interfaces within the network. The selection of the Grandmaster (GM), the primary reference clock, is determined by the Best Master Clock Algorithm (BMCA). Devices synchronize their clocks to the GM through a process known as the Delay Request-Response Mechanism (DRRM), wherein ports that are directly synchronizing with the GM enter a secondary state.

Historically, ports in a passive state—those that receive timing messages from FMs but aren't actively syncing to the GM—didn't participate in DRRM, which meant they didn't synchronize their clocks.

Starting Cisco IOS XR Software Release 24.2.1, DRRM has been extended to include passive ports, enabling them to engage in the exchange of delay request and response packets. This enhancement allows for the calculation of PTP phase differences between the clocks on passive ports and the GM.

This calculated PTP phase difference provides a valuable insight into the timing characteristics of other foreign masters in the network by using the grandmaster as a reference point. It can be utilized on any boundary clock or slave clock that has connections to at least one other foreign master.

You can access these measurements and the calculated PTP phase differences using show commands through the router's CLI. Also, the information can be retrieved programmatically through operational data models in YANG, providing flexibility in how you can access and utilize this synchronization data.

Phase Difference Alarm

PTP phase difference can also be used to monitor the timing properties of the network. You can configure a value at which a bistate alarm is triggered when the PTP phase difference of a FM exceeds the threshold. The PTP phase difference can have a negative or positive value, but the threshold can only be the absolute value. You can configure the PTP phase difference threshold using the **phase-difference-threshold-breach** command.

System Log for PTP Phase Difference

When the configured threshold is reached, system logs (syslogs) are displayed. The following syslog is triggered if the configured PTP phase difference threshold is passed through by any master.

```
Phase difference for clock ACDE48FFFE234567, steps removed 1, receiving-port 1, received
on interface GigabitEthernet0/2/0/3 is 40ns, configured threshold is 30ns. Raising phase
difference alarm.
```

Configure PTP Phase Difference Alarm Threshold

Step 1 Configure threshold for triggering PTP phase difference alarms using the **phase-difference-threshold-breach** command.

Example:

```
Router#configure terminal
Router(config)#ptp
Router(config-ptp)#phase-difference-threshold-breach 300
Router(config-ptp)#commit
```

Step 2 Verify that PTP phase difference threshold value is configured using the **show running configuration** command.

Example:

```
ptp
 phase-difference-threshold-breach 300
!
```

Step 3 Display the current operational value using the **show ptp foreign-masters** command.

Example:

```
Router#show ptp foreign-masters
Ethernet, Address 0102.0304.050a, Multicast
    Configured priority: 40
    Configured clock class: None
    Configured delay asymmetry: 3 microseconds
    Announce granted:   4 per-second,      600 seconds
    Sync granted:       4 per-second,      600 seconds
    Delay-resp granted: 4 per-second,      600 seconds
    Not qualified (PTSF lossSync)
    Clock ID: abcdef1
    Phase difference: 300ns
    Servo status: PTSF-unusable
    Received clock properties:
      Domain: 0, Priority1: 1, Priority2: 100, Class: 52
      Accuracy: 0x00, Offset scaled log variance: 0x0000
      Steps-removed: 2, Time source: GPS, Timescale: PTP
      Time-traceable
      Current UTC offset: 0 seconds
    Parent properties:
      Clock ID: 0
      Port number: 0
```

Isolate Foreign Masters Causing Packet Timing Signal Fail

Table 48: Feature History Table

Feature Name	Release Information	Feature Description
Isolate Foreign Masters Causing Packet Timing Signal Fail	Release 24.2.1	This feature permits the flexible selection of timing sources by filtering out Foreign Master (FM) clocks that exhibit unstable timing. This filtering causes the secondary clocks to produce a signal deemed Packet Timing Signal Fail (PTSF)-unusable, from consideration within the Best Master Clock Algorithm (BMCA). The system continuously monitors these clocks for timing stabilization, and upon detecting enhanced stability, it may reevaluate and possibly reintegrate them as suitable time sources. This feature introduces these changes: **CLI:** • **detect-ptsf-unusable** • The **show ptp foreign-masters** command output is enhanced to include phase difference values and servo status. **YANG Data Models:** The following data models are enhanced: • `Cisco-IOS-XR-ptp-cfg.yang` • `Cisco-IOS-XR-um-ptp-cfg.yang`

Starting Cisco IOS XR Software Release 24.2.1, the servo mechanism now has the ability to detect unusable clocks due to packet timing signal fail by analyzing timestamps from foreign masters. This enhancement allows the system to identify foreign masters with unstable timing as unsuitable for use. A platform supports multiple masters, such a master can be excluded from the BMCA selection process while remaining under observation for potential recovery. Even after a master is deemed unusable, the DRRM continues to operate and timestamps from it are still provided to the servo. This ongoing monitoring enables PTP to detect and respond to any improvements in the primary's timing, allowing it to be reconsidered as usable.

System Log for PTSF-unusable

When the master becomes PTSF-unusable, and if its the current Grandmaster, the following system log (syslogs) is displayed:

```
Foreign master with clock ID ACDE48FFFE234567, steps removed 1, receiving-port 1, received
 on interface GigabitEthernet0/2/0/4 is now PTSF-unusable and disqualified from selection.
```

Configure PTSF-unusable

Step 1 Exclude the FM with unstable timing from selection in the BMCA and declare it as unusable using the **detect-ptsf-unusable** command.

Example:

```
Router#configure terminal
Router(config)#ptp
Router(config-ptp)#detect-ptsf-unusable
Router(config-ptp)#commit
```

Step 2 Check if the master clock is PTSF-unuable using the **show ptp foreign-masters** command.

Example:

```
Router#show-ptp-foreign-masters
Ethernet, Address 0102.0304.050a, Multicast
    Configured priority: 40
    Configured clock class: None
    Configured delay asymmetry: 3 microseconds
    Announce granted:   4 per-second,      600 seconds
    Sync granted:       4 per-second,      600 seconds
    Delay-resp granted: 4 per-second,      600 seconds
    Not qualified (PTSF lossSync)
    Clock ID: abcdef1
    Phase difference: -5000ns
    Servo status: PTSF-unusable
    Received clock properties:
      Domain: 0, Priority1: 1, Priority2: 100, Class: 52
      Accuracy: 0x00, Offset scaled log variance: 0x0000
      Steps-removed: 2, Time source: GPS, Timescale: PTP
      Time-traceable
      Current UTC offset: 0 seconds
    Parent properties:
      Clock ID: 0
      Port number: 0
```

PTP Support Information

This table lists different types of support information related to PTP:

Transport Media	• UDP over IPv4 • Ethernet • IPv6

Messages	• Signaling • Announce • Sync • Follow-up • Delay-request • Delay-response • Management
Transport Modes	• Unicast: This is the default mode. All packets are sent as unicast messages. • Mixed: Announce and Sync messages are sent as multicast messages. Signaling, Delay-request, and Delay-response messages are sent as unicast messages. • Multicast: All packets are sent as multicast messages.

PTP Hardware Support Matrix

Table 49: Feature History Table

Feature Name	Release Information	Feature Description
Precision Time Protocol on 12-port 100 Gigabit Ethernet line cards, ASR 9000 5th generation 400G line cards, ASR 9902 Series Routers, and 0.8T PEC	Release 7.4.1	Support for IEEE-1588 PTP is extended to the following routers and line cards: • A99-12X100GE • A9K-4X100GE • ASR-9902 • A9K-8HG-FLEX-SE/TR • A9K-4HG-FLEX-SE • A9K-4HG-FLEX-TR • A99-4HG-FLEX-SE • A99-4HG-FLEX-TR

Feature Name	Release Information	Feature Description
PTP support on 5th Generation 10-Port 400 Gigabit Ethernet Line Cards: • A99-10X400GE-X-SE • A99-10X400GE-X-TR	Release 7.3.2	Support for IEEE-1588 PTP is extended to the following line cards: • A99-10X400GE-X-SE • A99-10X400GE-X-TR

Note The table also contains support details of upcoming releases. You can read this table in context of the current release and see relevant *Release Notes* for more information on supported features and hardware.

This table provides a detailed information on the supported hardware:

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A9K-8X100GE-L-SE/TR (10GE and 100GE)	Default & G.8265.1	5.3.3	6.3.2 6.4.1	PTP over Ethernet does not work on 100G ports on Cisco IOS XR until 6.4.1. Support was introduced in 6.4.1.
	G.8275.1 & G.8275.2	6.2.1	6.3.2 6.4.1	
	G.8273.2	6.2.1	6.3.2	
	PTP Multiprofile	6.5.1	6.5.1	
A9K-RSP880-SE/TR	1588/PTP Default & G.8265.1	5.3.3	6.3.2 6.4.1	-
	1588/PTP G.8275.1 & G.8275.2	6.2.1	6.3.2 6.4.1	
	1588/PTP G.8273.2	6.2.1	6.3.2 6.4.1	
	PTP Multiprofile	6.5.1	6.5.1	

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A9K-8X100GE-L-SE/TR (40-GE)	1588/PTP Default & G.8265.1	6.0.1	6.3.2 6.4.1	-
	1588/PTP G.8275.1 & G.8275.2	6.2.1	6.3.2 6.4.1	
	1588/PTP G.8273.2	NA	NA	
	PTP Multiprofile	6.5.1	6.5.1	
A9K-4X100GE-SE/TR A9K-8X100GE-SE/TR	1588/PTP Default & G.8265.1	6.2.1	6.4.1	PTP over Ethernet does not work on 100G ports on Cisco IOS XR until 6.4.1. Support was introduced in 6.4.1. In 6.2.1, only G.8275.1 PTP profile is supported on the cards; No support for G.8273.2 PTP profile.
	1588/PTP G.8275.1 & G.8275.2	6.2.1	6.4.1	
	1588/PTP G.8273.2	6.4.1	6.4.1	
	PTP Multiprofile	6.5.1	6.5.1	
A9K-MOD400-SE/TR & A9K-MOD200-SE/TR with Legacy MPAs	1588/PTP Default & G.8265.1	6.1.3	6.4.1	-
	1588/PTP G.8275.1 & G.8275.2	6.2.2	6.4.1	-
	1588/PTP G.8273.2	-	-	-
	PTP Multiprofile	6.5.1	6.5.1	-

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A9K-MOD400-SE/TR & A9K-MOD200-SE/TR with MPA 20x10GE , A9K-MPA-1X100GE and A9K-MPA-2X100GE	1588/PTP Default & G.8265.1	6.1.3	6.4.1	PTP over Ethernet does not work on 100G ports on Cisco IOS XR until 6.4.1. Support was introduced in 6.4.1. In 6.2.2, only G.8275.1 PTP profile is supported on the cards. No support for G.8273.2 PTP profile until 6.5.1.
	1588/PTP G.8275.1 & G.8275.2	6.2.2	6.4.1	
	1588/PTP G.8273.2	6.5.1	6.5.1	
	PTP Multiprofile	6.5.1	6.5.1	
A9K-24X10GE-1G-SE/TR A9K-48X10GE-1G-SE/TR	1588/PTP Default & G.8265.1	6.2.2 6.3.1	6.3.2	-
	1588/PTP G.8275.1 & G.8275.2	6.2.2 6.3.1	6.3.2	
	1588/PTP G.8273.2	6.3.1	6.3.2	
	PTP Multiprofile	6.5.1	6.5.1	
A99-RSP-SE/TR (Cisco ASR 9910 Series Routers)	1588/PTP Default & G.8265.1	6.3.1	6.3.2	-
	1588/PTP G.8275.1 & G.8275.2	6.3.1	6.3.2	
	1588/PTP G.8273.2	6.4.1	6.3.2	
	PTP Multiprofile	6.5.1	6.5.1	

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A9K-RSP880-LT-SE/TR	1588/PTP Default & G.8265.1	6.2.2	6.4.1	-
	1588/PTP G.8275.1 & G.8275.2	6.2.2	6.4.1	
	1588/PTP G.8273.2	6.4.1	6.4.1	
	PTP Multiprofile	6.5.1	6.5.1	
A9K-RSP440-TR/SE A99-RP-SE Enhanced Ethernet Linecards	1588/PTP Default & G.8265.1	4.3.4	NA	Enhanced Ethernet linecards do not support G.8273.2 with G.8275.1 PTP profile. .
	1588/PTP G.8275.1 & G.8275.2	NA	NA	
	1588/PTP G.8273.2	NA	NA	
A99-RP2-TR/SE	1588/PTP Default & G.8265.1	5.3.3	6.3.2 6.4.1	-
	1588/PTP G.8275.1 & G.8275.2	6.2.1	6.3.2 6.4.1	
	1588/PTP G.8273.2	NA	NA	
Cisco ASR 9001 Series Routers	1588/PTP Default & G.8265.1	4.3.4	NA	Enhanced Ethernet based hardware does not support G.8273.2 with G.8275.1 PTP profile.
	1588/PTP G.8275.1 & G.8275.2	NA	NA	
	1588/PTP G.8273.2	NA	NA	

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
Cisco ASR 9901 Series Routers	1588/PTP Default & G.8265.1	NA	6.4.1	-
	1588/PTP G.8275.1 & G.8275.2	NA	6.4.1	
	1588/PTP G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.5.1	
A99-RSP-SE/TR (Cisco ASR 9906 Series Routers)	1588/PTP Default & G.8265.1	6.3.1	6.3.2	-
	1588/PTP G.8275.1 & G.8275.2	6.3.1	6.3.2	
	1588/PTP G.8273.2	6.4.1	6.3.2	
	PTP Multiprofile	6.5.1	6.5.1	
A9K-RSP5-SE	1588/PTP Default & G.8265.1	NA	6.5.15	-
	1588/PTP G.8275.2	NA	6.5.15	
	1588/PTP G.8275.1 & G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.5.15	
A9K-RSP5-TR	1588/PTP Default & G.8265.1	NA	6.5.15	-
	1588/PTP G.8275.2	NA	6.5.15	
	1588/PTP G.8275.1 & G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.5.15	

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A99-RP3-SE	1588/PTP Default & G.8265.1	NA	6.5.15	-
	1588/PTP G.8275.2	NA	6.5.15	
	1588/PTP G.8275.1 & G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.5.15	
A99-RP3-TR	1588/PTP Default & G.8265.1	NA	6.5.15	-
	1588/PTP G.8275.2	NA	6.5.15	
	1588/PTP G.8275.1 & G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.5.15	
A9K-8X100GE-X-TR	1588/PTP Default & G.8265.1	NA	6.5.15	-
	1588/PTP G.8275.2	NA	6.5.15	
	1588/PTP G.8275.1 & G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.5.15	
A9K-16X100GE-TR	1588/PTP Default & G.8265.1	NA	6.5.15	NA
	1588/PTP G.8275.2	NA	6.5.15	
	1588/PTP G.8275.1 & G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.5.15	
A99-16X100GE-X-SE A99-32X100GE-TR/CM	1588/PTP Default & G.8265.1	NA	6.6.1	NA
	1588/PTP G.8275.2	NA	6.6.1	
	1588/PTP G.8275.1 & G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.6.1	

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A9K-32X100GE-TR	1588/PTP Default & G.8265.1	NA	6.5.15	-
	1588/PTP G.8275.2	NA	6.5.15	
	1588/PTP G.8275.1 & G.8273.2	NA	6.6.1	
	PTP Multiprofile	NA	6.5.15	
Cisco ASR 9903 Series Routers	1588/PTP Default & G.8265.1	NA	7.1.3	You must configure *'one-step'* clock operation on the *PTP master interface.*
	1588/PTP G.8275.2	NA	7.1.3	
	1588/PTP G.8275.1 & G.8273.2	NA	7.1.3	
	PTP Multiprofile	NA	NA	
A9903-20HG-PEC	1588/PTP Default & G.8265.1	NA	7.1.3	
	1588/PTP G.8275.2	NA	7.1.3	
	1588/PTP G.8275.1 & G.8273.2	NA	7.1.3	
	PTP Multiprofile	NA	NA	
A99-32X100GE-X-SE/TR	1588/PTP Default & G.8265.1	NA	7.1.15	
	1588/PTP G.8275.2	NA	7.1.15	
	1588/PTP G.8275.1 & G.8273.2	NA	7.1.15	
	PTP Multiprofile	NA	NA	
A9K-8HG-FLEX-SE/TR	1588/PTP Default & G.8265.1	NA	7.1.15	You must configure *'one-step'* clock operation on the *PTP master interface.*
	1588/PTP G.8275.2	NA	7.1.15	
	1588/PTP G.8275.1 & G.8273.2	NA	7.1.15	
	PTP Multiprofile	NA	NA	

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A9K-20HG-FLEX-SE/TR	1588/PTP Default & G.8265.1	NA	7.1.15	You must configure *'one-step'* clock operation on the *PTP master interface.*
	1588/PTP G.8275.2	NA	7.1.15	
	1588/PTP G.8275.1 & G.8273.2	NA	7.1.15	
	PTP Multiprofile	NA	NA	
A99-10X400GE-X-SE/TR	1588/PTP Default & G.8265.1	NA	7.3.2	You must configure *'one-step'* clock operation on the *PTP master interface.* Class B Performance (Applicable to 1588/PTP G.8275.1 & G.8273.2)
	1588/PTP G.8275.2	NA	7.3.2	
	1588/PTP G.8275.1 & G.8273.2	NA	7.3.2	
	PTP Multiprofile	NA	NA	
A99-12x100GE A99-12X100GE-CM	1588/PTP Default & G.8265.1	NA	7.4.1	
	1588/PTP G.8275.2	NA	7.4.1	
	1588/PTP G.8275.1 & G.8273.2	NA	7.4.1	Class B Performance
	PTP Multiprofile	NA	7.4.1	
A99-8X100GE-SE/TR/CM A9K-8X100GE-CM A9K-8X100G-LB-SE/TR A9K-400G-DWDM-TR A99-48X10GE-1G-SE/TR	1588/PTP Default & G.8265.1	6.2.2	6.2.2	
	1588/PTP G.8275.2	6.2.2	6.2.2	
	1588/PTP G.8275.1 & G.8273.2	6.2.2	6.2.2	
	PTP Multiprofile	6.2.2	6.2.2	
A9K-4X100GE	1588/PTP Default & G.8265.1	NA	7.4.1	
	1588/PTP G.8275.2	NA	7.4.1	
	1588/PTP G.8275.1 & G.8273.2	NA	7.4.1	Class B Performance
	PTP Multiprofile	NA	NA	

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A9K-400GE-SE/TR	1588/PTP Default & G.8265.1	NA	7.4.1	
	1588/PTP G.8275.2	NA	7.4.1	
	1588/PTP G.8275.1 & G.8273.2	NA	7.4.1	Class B Performance
	PTP Multiprofile	NA	NA	
A99-400GE-SE/TR	1588/PTP Default & G.8265.1	NA	7.4.1	
	1588/PTP G.8275.2	NA	7.4.1	
	1588/PTP G.8275.1 & G.8273.2	NA	7.4.1	Class B Performance
	PTP Multiprofile	NA	NA	
ASR 9902	1588/PTP Default & G.8265.1	NA	7.4.1	Port 12 to Port 35 provides Class B Performance and requires two-step clock operation on PTP master interface Port 0 to port 11 and port 36 to port 47 provide Class C performance and requires one-step clock operation on PTP master interface.
	1588/PTP G.8275.2	NA	7.4.1	
	1588/PTP G.8275.1 & G.8273.2	NA	7.4.1	
	PTP Multiprofile	NA	NA	
ASR-9903	1588/PTP Default & G.8265.1	NA	7.4.1	You must configure *'one-step'* clock operation on the *PTP master interface.*
	1588/PTP G.8275.2	NA	7.4.1	
	1588/PTP G.8275.1 & G.8273.2	NA	7.4.1	
	PTP Multiprofile	NA	NA	

Hardware Variant	1588/PTP	Cisco IOS XR	Cisco IOS XR 64 bit	Comments
A9K-4HG-FLEX-SE/TR	1588/PTP Default & G.8265.1	NA	7.4.1	You must configure *'one-step'* clock operation on the *PTP master interface.*
	1588/PTP G.8275.2	NA	7.4.1	
	1588/PTP G.8275.1 & G.8273.2	NA	7.4.1	
	PTP Multiprofile	NA	7.4.1	
A99-4HG-FLEX-SE/TR	1588/PTP Default & G.8265.1	NA	7.4.1	You must configure *'one-step'* clock operation on the *PTP master interface.*
	1588/PTP G.8275.2	NA	7.4.1	
	1588/PTP G.8275.1 & G.8273.2	NA	7.4.1	
	PTP Multiprofile	NA	7.4.1	

Note The following 2nd generation line cards support all IEEE-1588 PTP telecom profiles (Default, G.8265.1, G.8275.2, G.8275.1, G.8273.2, and PTP Multiprofile) in Cisco IOS XR 32 bit:

Table 50: 2nd Generation Line Cards Supporting IEEE-1588 PTP

Hardware Variant	Hardware Variant	Hardware Variant
A9K-2X100GE-SE/TR	A9K-40GE-SE/TR	A9K-40GE-SE/TR
A9K-1X100GE-SE/TR	A9K-MOD160-SE/TR	A9K-VSM-500
A9K-36X10GE-SE/TR	A9K-MOD80-SE/TR	A9K-SIP-700
A9K-4T16GE-SE/TR	A9K-4T16GE-SE/TR	

Restrictions

- PTP Grandmaster (GM) is not supported with all the PTP profiles.
- RSP IEEE 1588 port on RSP/RP is not supported.
- Two-step clock operation is recommended over one-step clock operation for a PTP server.
- If PTP clock operation CLI is not configured, the default clock operation is two-step on all ASR9000 hardware variants.
- Due to the difference in PTP timestamp unit, which involves the PHY injecting the timestamp instead of the NPU, you must configure PTP clock operation **one-step** on the PTP master interface of the line

cards which are explicitly specified in the PTP Hardware Support Matrix, on page 368. Rest of the line cards only support PTP clock operation **two-step** on the PTP master interface.

- PTP clock operation one-step or two-step restriction is only for PTP master interface. PTP slave interface can operate in either one-step or two-step.
- Cisco ASR 9000 Series Routers do not support Class B 1 Pulse Per Second (PPS) performance with Forward Error Correction (FEC) enabled optics.
- The upgrade of TimingIC-*X* firmware impacts timing functionality on the ASR 9902 and ASR 9903 Routers, and 5th Generation Line Cards. You must reload the linecard after upgrade of timing firmware for proper functioning of SyncE and PTP features. Interface flapping and traffic drops are expected during this process.
- G.8275.1 and G.8275.2 profiles are not supported on Cisco ASR 9001 chassis, Cisco ASR 9000 Ethernet line cards, Cisco ASR 9000 Enhanced Ethernet line cards, and A9K-400G-DWDM-SE/TR line cards.
- As recommended in Appendix VI of ITU-T G.8275.1 document, G.8275.1 profile is supported only on Bundle Link Aggregation (LAG) member links and not supported on a bundle interface.
- G.8273.2 Telecom Boundary Clock (T-BC) performance is not supported on 40G interfaces.
- The G.8273.2 Class B performance is observed when the same type of line card is used for both PTP server and PTP client ports. Class A performance is observed when different types of line cards are used for PTP server and PTP client on T-BC.
- G.8275.2 profile is supported on Cisco ASR 9000 Series Routers. However, the performance standards of this profile are not aligned with any of the ITU-T standards because performance specifications for G.8275.2 profile has not yet been made available by ITU-T.
- Transparent Clock (TC) is not supported.
- PTP Multiprofile is not supported for G.8273.2 Class B performance.
- Platform Fault Manager (PFM) alarms for the 10MHz port are not supported on A9K-RSP5-SE, A9K-RSP5-TR, A99-RP3-SE, and A99-RP3-TR.
- Select 5th generation line cards (A9K-20HG-FLEX-xx and A9K-8HG-FLEX-xx) will support PTP Telecom Profile G.8275.2 in combination with transit G.8265.1/G.8275.2 packets, in a future version of these cards.

Note Forwarding PTP packets as IP or MPLS isn't possible without the redirecting device not being PTP-aware. If each node across the PTP path isn't performing the T-BC function, timing accuracy can't be maintained.

ITU-T Telecom Profiles for PTP

Cisco IOS XR software supports ITU-T Telecom Profiles for PTP as defined in the ITU-T recommendation. A profile consists of PTP configuration options applicable only to a specific application.

Separate profiles can be defined to incorporate PTP in different scenarios based on the IEEE 1588-2008 standard. A telecom profile differs in several ways from the default behavior defined in the IEEE 1588-2008 standard and the key differences are mentioned in the subsequent sections.

The following sections describe the ITU-T Telecom Profiles that are supported for PTP.

G.8265.1 Profile

G.8265.1 profile fulfills specific frequency-distribution requirements in telecom networks. Features of G.8265.1 profile are:

- *Clock advertisement*: G.8265.1 profile specifies changes to values used in Announce messages for advertising PTP clocks. The clock class value is used to advertise the quality level of the clock, while the other values are not used.
- *Clock Selection*: G.8265.1 profile also defines an alternate Best Master Clock Algorithm (BMCA) to select port states and clocks is defined for the profile. This profile also requires to receive Sync messages (and optionally, Delay-Response messages) to qualify a clock for selection.
- *Port State Decision*: The ports are statically configured to be Master or Slave instead of using FSM to dynamically set port states.
- *Packet Rates*: The packet rates higher than rates specified in the IEEE 1588-2008 standard are used. They are:
  - Sync/Follow-Up Packets: Rates from 128 packets-per-second to 16 seconds-per-packet.
  - Delay-Request/Delay-Response Packets: Rates from 128 packets-per-second to 16 seconds-per-packet.
  - Announce Packets: Rates from 8 packets-per-second to 64 packets-per-second.
- *Transport Mechanism*: G.8265.1 profile only supports IPv4 PTP transport mechanism.
- *Mode*: G.8265.1 profile supports transport of data packets only in unicast mode.
- *Clock Type*: G.8265.1 profile only supports Ordinary Clock-type (a clock with only one PTP port).
- *Domain Numbers*: The domain numbers that can be used in a G.8265.1 profile network ranges from 4 to 23. The default domain number is 4.
- *Port Numbers*: All PTP port numbers can only be 1 because all clocks in a this profile network are Ordinary Clocks.

G.8265.1 profile defines an alternate algorithm to select between different master clocks based on the local priority given to each master clock and their quality levels (QL). This profile also defines Packet Timing Signal Fail (PTSF) conditions to identify the master clocks that do not qualify for selection. They are:

- PTSF-lossSync condition: Raised for master clocks that do not receive a reliable stream of Sync and Delay-Resp messages. Cisco IOS XR software requests Sync and Delay-Resp grants for each configured master clock to track the master clock with this condition.
- PTSF-lossAnnounce condition: Raised for master clocks that do not receive a reliable stream of Announce messages.
- PTSF-unusable condition: Raised for master clocks that receives a reliable stream of Announce, Sync, and Delay-Resp messages, but not usable by slave clocks. Cisco IOS XR software does not use this condition.

Hardware variant-specific behavior

The profile G8265.1 displays the following behavior on these hardware variants A9K-RSP5-SE, A9K-RSP5-TR, A99-RP3-SE, and A99-RP3-TR:

- Configuring either a master or slave clock type is mandatory.
- G.8265.1 is only a frequency synchronization profile and the servo state is displayed as FREQ_LOCKED and the PTP slave interface remains as slave. Phase synchronization is not supported.
- G.8265.1 profile supports only PTP pure mode and not PTP hybrid mode.

G.8275.1 Profile

G.8275.1 profile fulfills the time-of-day and phase synchronization requirements in telecom networks with all network devices participating in the PTP protocol. G.8275.1 profile with SyncE provides better frequency stability for the time-of-day and phase synchronization.

Features of G.8275.1 profile are:

- *Synchronization Model*: G.8275.1 profile adopts hop-by-hop synchronization model. Each network device in the path from master to slave synchronizes its local clock to upstream devices and provides synchronization to downstream devices.
- *Clock Selection*: G.8275.1 profile also defines an alternate BMCA that selects a clock for synchronization and port state for the local ports of all devices in the network is defined for the profile. The parameters defined as a part of the BMCA are:
  - Clock Class
  - Clock Accuracy
  - Offset Scaled Log Variance
  - Priority 2
  - Clock Identity
  - Steps Removed
  - Port Identity
  - notSlave flag
  - Local Priority
- *Port State Decision*: The port states are selected based on the alternate BMCA algorithm. A port is configured to a **master-only** port state to enforce the port to be a master for multicast transport mode.
- *Packet Rates*: The nominal packet rate for Announce packets is 8 packets-per-second and 16 packets-per-second for Sync/Follow-Up and Delay-Request/Delay-Response packets.
- *Transport Mechanism*: G.8275.1 profile only supports Ethernet PTP transport mechanism.
- *Mode*: G.8275.1 profile supports transport of data packets only in multicast mode. The forwarding is done based on forwardable or non-forwardable multicast MAC address.
- *Clock Type*: G.8275.1 profile supports the following clock types:

  - *Telecom Grandmaster (T-GM)*: Provides timing for other network devices and does not synchronize its local clock to other network devices.
  - *Telecom Time Slave Clock (T-TSC)*: A slave clock synchronizes its local clock to another PTP clock, but does not provide PTP synchronization to any other network devices.
  - *Telecom Boundary Clock (T-BC)*: Synchronizes its local clock to a T-GM or an upstream T-BC clock and provides timing information to downstream T-BC or T-TSC clocks.

- *Domain Numbers*: The domain numbers that can be used in a G.8275.1 profile network ranges from 24 to 43. The default domain number is 24.

Hardware variant-specific behavior

The profile G8275.1 displays the following behavior on these hardware variants A9K-RSP5-SE, A9K-RSP5-TR, A99-RP3-SE, and A99-RP3-TR:

- SyncE input is mandatory as only PTP hybrid mode is supported.
- The frequency is derived from the SyncE interface and phase adjustments are based on PTP.
- If you configure SyncE before you configure PTP, the Servo state is set to FREQ_LOCKED by default.
- After the Servo is in PHASE_LOCKED state, if the SyncE input is lost or removed, the Servo transitions to HOLDOVER state.
- After the Servo is in PHASE_LOCKED state, if the PTP input is lost or removed, the Servo transitions to FREQ_LOCKED state.

Note For the hardware variants A9K-8X100GE-X-TR, A9K-16X100GE-TR and A9K-32X100GE-TR you are not required to shut the 100 GE link to configure this profile.

G.8275.2 Profile

G.8275.2 profile fulfills the time-of-day and phase synchronization requirements in telecom networks with partial timing support from the network. Features of G.8275.2 profile are:

- *Clock Selection*: G.8275.2 profile also defines an alternate BMCA that selects a clock for synchronization and port state for the local ports of all devices in the network is defined for the profile. The parameters defined as a part of the BMCA are:
  - Clock Class
  - Clock Accuracy
  - Offset Scaled Log Variance
  - Priority 2
  - Clock Identity
  - Steps Removed
  - Port Identity

- notSlave flag
- Local Priority

Note See ITU-T G.8275.2 document to determine the valid values for Clock Class parameter.

- *Port State Decision*: The port states are selected based on the alternate BMCA algorithm. A port is configured to a **master-only** port state to enforce the port to be a master for unicast transport mode.
- *Packet Rates:*
    - Synchronization/Follow-Up—minimum is one packet-per-second and maximum of 128 packets-per-second.
    - Packet rate for Announce packets—minimum of one packet-per-second and maximum of eight packets-per-second.
    - Delay-Request/Delay-Response packets—minimum is one packet-per-second and maximum of 128 packets-per-second
- *Transport Mechanism*: G.8275.2 profile supports only IPv4 and IPv6 PTP transport mechanism.
- *Mode*: G.8275.2 profile supports transport of data packets only in unicast mode.
- *Clock Type*: G.8275.2 profile supports the following clock types:
    - *Telecom Grandmaster (T-GM)*: Provides timing for other network devices and does not synchronize its local clock to other network devices.
    - *Telecom Time Slave Clock (T-TSC)*: A slave clock synchronizes its local clock to another PTP clock, but does not provide PTP synchronization to any other network devices.
    - *Telecom Boundary Clock (T-BC)*: Synchronizes its local clock to a T-GM or an upstream T-BC clock and provides timing information to downstream T-BC or T-TSC clocks.
- *Domain Numbers*: The domain numbers that can be used in a G.8275.2 profile network ranges from 44 to 63. The default domain number is 44.

Hardware variant-specific behavior

The profile G8275.2 displays the following behavior on these hardware variants A9K-RSP5-SE, A9K-RSP5-TR, A99-RP3-SE, and A99-RP3-TR:

- Hybrid PTP and pure PTP are supported on this profile.
- The physical-layer-frequency command must be used to configure Hybrid PTP.
- To switch from Hybrid PTP to Pure PTP, you must remove the physical-layer-frequency configuration and frequency synchronization configuration to remove SyncE inputs from line card interfaces and RSP clock-interfaces.

Configuring PTP

Prerequisite

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

PTP Interface and Profile Configuration

When a global PTP profile is attached to an interface, its values are used as default settings for that interface. When additional settings are configured under an interface itself, these settings override the defaults in that profile. When no profile is attached to an interface, the configuration on the interface is used to determine the PTP settings for that interface.

When configuring PTP, use one of the following approaches:

- Create a profile (or multiple profiles) containing all the default settings to use on all PTP interfaces. Override any settings that differ for particular interfaces by using the interface configuration under the interfaces themselves.
- Configure all settings separately for each interface, without using any global profiles. Use this approach if the interfaces do not have consistent settings, or if you are configuring only a small number of PTP interfaces.

Configuring Frequency Synchronization and Quality Settings for PTP

This procedure describes the steps involved to configure frequecy and quality settings for PTP on a router.

1. To enable frequency synchronization on the router, use **frequency synchronization** command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# frequency synchronization
```

2. To configure ITU-T quality parameters, use **quality itu-t option** *option* **generation** *number* command in the frequency synchronization configuration mode.
   - **option 1**: Includes PRC, SSU-A, SSU-B, SEC, and DNU. This is the default option.
   - **option 2 generation 1**: Includes PRS, STU, ST2, ST3, SMC, and DUS.
   - **option 2 generation 2**: Includes PRS, STU, ST2, ST3, TNC, ST3E, SMC, and DUS.

Note The **quality option** configured here must match the **quality option** specified in the **quality receive** and **quality transmit** commands.

```
RP/0/RSP0/CPU0:router(config-freqsync)# quality itu-t
option 2 generation 2
```

Verification

To display the frequency synchronization selection, use **show frequency synchronization selection** command.

```
RP/0/RSP0/CPU0:router# show frequency synchronization selection
Node 0/RSP1/CPU0:
==============
Selection point: T0-SEL-B (3 inputs, 1 selected)
  Last programmed 06:49:27 ago, and selection made 06:49:15 ago
  Next selection points
    SPA scoped    : None
    Node scoped   : T4-SEL-C CHASSIS-TOD-SEL
    Chassis scoped: LC_TX_SELECT
    Router scoped : None
  Uses frequency selection
  Used for local line interface output
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
  1  Sync1 [0/RSP1/CPU0]       n/a                         PRC    1  Locked
     HundredGigE0/5/0/2        0/5/CPU0 ETH_RXMUX 1        PRC    1  Available
     Internal0 [0/RSP1/CPU0]   n/a                         SEC  255  Available

Selection point: T4-SEL-A (1 inputs, 1 selected)
  Last programmed 06:49:27 ago, and selection made 06:49:15 ago
  Next selection points
    SPA scoped    : None
    Node scoped   : T4-SEL-C
    Chassis scoped: None
    Router scoped : None
  Uses frequency selection
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
  1  HundredGigE0/5/0/2        0/5/CPU0 ETH_RXMUX 1        PRC    1  Available

Selection point: T4-SEL-C (2 inputs, 1 selected)
  Last programmed 06:49:15 ago, and selection made 06:49:15 ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: None
    Router scoped : None
  Uses frequency selection
  Used for local clock interface output
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
  1  Sync1 [0/RSP1/CPU0]       0/RSP1/CPU0 T0-SEL-B 1      PRC    1  Locked
     HundredGigE0/5/0/2        0/RSP1/CPU0 T4-SEL-A 1      PRC    1  Available

Selection point: CHASSIS-TOD-SEL (1 inputs, 1 selected)
  Last programmed 6d04h ago, and selection made 6d04h ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: None
    Router scoped : None
  Uses time-of-day selection
  S  Input                     Last Selection Point      Pri  Time  Status
  == ========================  ========================  ===  ====  ===========
  1  Sync1 [0/RSP1/CPU0]       0/RSP1/CPU0 T0-SEL-B 1    100  Yes   Available

Node 0/3/CPU0:
==============
Selection point: ETH_RXMUX (0 inputs, 0 selected)
```

```
  Last programmed 9w6d ago, and selection made 9w6d ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: T0-SEL-B T4-SEL-A
    Router scoped : None
  Uses frequency selection

Selection point: LC_TX_SELECT (1 inputs, 1 selected)
  Last programmed 9w6d ago, and selection made 9w6d ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: None
    Router scoped : None
  Uses frequency selection
  Used for local line interface output
  S  Input                    Last Selection Point        QL  Pri  Status
  == ======================== ======================== ===== ===  ===========
  24 Sync1 [0/RSP1/CPU0]       0/RSP1/CPU0 T0-SEL-B 1      PRC    1  Available

Node 0/5/CPU0:
==============
Selection point: ETH_RXMUX (1 inputs, 1 selected)
  Last programmed 06:49:27 ago, and selection made 06:49:27 ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: T0-SEL-B T4-SEL-A
    Router scoped : None
  Uses frequency selection
  S  Input                    Last Selection Point        QL  Pri  Status
  == ======================== ======================== ===== ===  ===========
  1  HundredGigE0/5/0/2        n/a                         PRC    1  Available

Selection point: LC_TX_SELECT (1 inputs, 1 selected)
  Last programmed 6d04h ago, and selection made 6d04h ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: None
    Router scoped : None
  Uses frequency selection
  Used for local line interface output
  S  Input                    Last Selection Point        QL  Pri  Status
  == ======================== ======================== ===== ===  ===========
  24 Sync1 [0/RSP1/CPU0]       0/RSP1/CPU0 T0-SEL-B 1      PRC    1  Available
```

Configuring Global Profile

This procedure describes the steps involved to create a global configuration profile for a PTP interface that can then be assigned to any interface as required.

Note Prior to Cisco IOS XR Software Release 6.3.3, the default PTP timers for G2875.1 were not set to standard values. This could lead to interoperability issues with other routers running the timers with updated values. Hence, to prevent such issues arising due to difference in packet rates, you must explicitly configure the **announce interval** value to 8, **sync frequency** value to 16 and **delay-request frequency** value to 16 while configuring global g.2875.1 profile.

1. To enter the PTP configuration mode, use **ptp** command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# ptp
```

2. To configure a PTP profile, use **profile** command in the ptp configuration mode.

```
RP/0/RSP0/CPU0:router(config-ptp)# profile tp64
```

3. To configure frequency for a Sync message for the given PTP profile, use **sync frequency** *rate* command in the ptp-profile configuration mode.

```
RP/0/RSP0/CPU0:router(config-ptp-profile)# sync frequency 16
```

4. To configure delay-request frequency for the given PTP profile, use **delay-request frequency** *rate* command in the ptp-profile configuration mode.

```
RP/0/RSP0/CPU0:router(config-ptp-profile)# delay-request frequency 16
```

Verification

To display the configured PTP profile details, use **show run ptp** command.

```
RP/0/RSP0/CPU0:router# show run ptp

Wed Feb 28 11:16:05.943 UTC
ptp
clock
  domain 24
  profile g.8275.1 clock-type T-BC
!
profile slave
  transport ethernet
  sync frequency 16
  announce interval 1
  delay-request frequency 16
!
profile master
  transport ethernet
  sync frequency 16
  announce interval 1
  delay-request frequency 16
!
profile slave1
  transport ethernet
  sync frequency 64
  announce interval 1
  delay-request frequency 64
```

```
!
```

Configuring PTP Slave Interface

This procedure describes the steps involved to configure a PTP interface to be a Slave.

1. To configure an interface, use **interface** *type interface-path-id* command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/5
```

2. To enter the PTP configuration mode for the given interface, use **ptp** command in the interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if)# ptp
```

3. To configure a PTP profile (or specify a previously defined profile), use **profile** *name* command in the ptp interface configuration mode.

Note Any additional commands entered in ptp-interface configuration mode overrides the global profile settings.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# profile tp64
```

4. To configure the transport mode for all PTP messages in the given PTP profile, use **transport** *mode_type* command in the ptp interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# transport ipv4
```

5. To configure timeout for PTP announce messages in the given PTP profile, use **announce interval** *interval-value* command in the ptp interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# announce interval 1
```

6. To configure the port state, use **port state** command in the ptp interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# port state slave-only
```

7. To configure IPv4 or IPv6 address for PTP master, use **master ipv4|ipv6** *address* command in the ptp interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# master ipv4 192.168.2.1
```

```
RP/0/RSP0/CPU0:router(config-if-ptp)# master ipv6 2001:DB8::1
```

8. To return to the interface configuration mode, use **exit** command.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# exit
```

9. To configure a gateway for the given interface, use **ipv4 address** *address mask* command in the interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 1.7.1.2 255.255.255.0
```

Verification

To verify the port state details, use **show run interface** *interface-name* command.

```
RP/0/RSP0/CPU0:router# show run interface TenGigE 0/1/0/5

Fri Aug  3 19:57:14.184 UTC
interface TenGigE 0/1/0/5
 ptp
  profile tp64
  transport ipv4
  port state slave-only
  master ipv4 192.168.2.1
  !
  announce interval 1
  !
 ipv4 address 1.7.1.1 255.255.255.0
!
```

Configuring PTP Master Interface

This procedure describes the steps involved to configure a PTP interface to be a Master.

1. To configure an interface, use **interface** *type interface-path-id* command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/5
```

2. To enter the PTP configuration mode for the given interface, use **ptp** command in the interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if)# ptp
```

3. To configure a PTP profile (or specify a previously defined profile), use **profile** *name* command in the ptp interface configuration mode.

Note Any additional commands entered in PTP interface configuration mode override settings in this profile.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# profile tp64
```

4. To configure the transport mode for all PTP messages in the given PTP profile, use **transport** *mode_type* command in the ptp interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# transport ipv4
```

5. To configure timeout for PTP announce messages in the given PTP profile, use **announce interval** *interval-value* command in the ptp interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# announce interval 1
```

6. To return to the interface configuration mode, use **exit** command.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# exit
```

7. To configure a gateway for the given interface, use **ipv4 address** *address mask* command in the interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 1.7.1.2 255.255.255.0
```

Verification

To verify the port state details, use **show run interface** *interface-name* command.

```
RP/0/RSP0/CPU0:router# show run interface TenGigE 0/1/0/5

Fri Aug  3 13:57:44.366 PST
interface TenGigE 0/1/0/5
 ptp
  profile tp64
  transport ipv4
  !
  announce interval 1
  !
  ipv4 address 1.7.1.2 255.255.255.0
 !
```

Configuring PTP Hybrid Mode

This procedure describes the steps involved to configure router in a hybrid mode. You can do this by selecting PTP for Time-of-Day (ToD) and another source for frequency.

1. To enable frequency synchronization on the router, use **frequency synchronization** command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# frequency synchronization
```

2. To configure a SyncE source, create an interface to be a SyncE input. This can be configured using **interface** command in the configuration mode.

Note The time-of-day-priority setting specifies that SyncE to be used as a ToD source if there is no source available with a lower priority.

```
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/1/0/0
RP/0/RSP0/CPU0:router(config-if)# frequency synchronization
RP/0/RSP0/CPU0:router(config-if-freqsync)# selection input
RP/0/RSP0/CPU0:router(config-if-freqsync)# time-of-day-priority 100
RP/0/RSP0/CPU0:router(config-if-freqsync)# commit
```

3. To configure PTP as the source for ToD, enable PTP on the router using **ptp** command in command in the configuration mode. ToD priority values can range from 1 (highest priority) to 254 (lowest priority).

```
RP/0/RSP0/CPU0:router(config)# ptp
RP/0/RSP0/CPU0:router(config-ptp)# time-of-day-priority 1
RP/0/RSP0/CPU0:router(config)# commit
```

4. To confiure a PTP interface, use **interface** command in configuration mode. To enable this interface as a PTP Master, use **master** command in ptp-interface configuration mode.

```
RP/0/RSP0/CPU0:router(config)# interface gigabitEthernet 0/1/0/1
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.0.0.1/24
RP/0/RSP0/CPU0:router(config-if)# ptp
RP/0/RSP0/CPU0:router(config-if-ptp)# master ipv4 10.0.0.2
RP/0/RSP0/CPU0:router(config-if-ptp)# commit
```

Verification

To display the frequency synchronization selection, use **show frequency synchronization selection** command.

```
RP/0/RSP0/CPU0:router# show frequency synchronization selection
Node 0/RSP1/CPU0:
==============
Selection point: T0-SEL-B (3 inputs, 1 selected)
  Last programmed 06:49:27 ago, and selection made 06:49:15 ago
  Next selection points
    SPA scoped    : None
    Node scoped   : T4-SEL-C CHASSIS-TOD-SEL
    Chassis scoped: LC_TX_SELECT
    Router scoped : None
  Uses frequency selection
  Used for local line interface output
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
  1  Sync1 [0/RSP1/CPU0]       n/a                         PRC    1  Locked
     HundredGigE0/5/0/2        0/5/CPU0 ETH_RXMUX 1        PRC    1  Available
     Internal0 [0/RSP1/CPU0]   n/a                         SEC  255  Available

Selection point: T4-SEL-A (1 inputs, 1 selected)
  Last programmed 06:49:27 ago, and selection made 06:49:15 ago
  Next selection points
    SPA scoped    : None
    Node scoped   : T4-SEL-C
    Chassis scoped: None
    Router scoped : None
  Uses frequency selection
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
  1  HundredGigE0/5/0/2        0/5/CPU0 ETH_RXMUX 1        PRC    1  Available

Selection point: T4-SEL-C (2 inputs, 1 selected)
  Last programmed 06:49:15 ago, and selection made 06:49:15 ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: None
    Router scoped : None
  Uses frequency selection
  Used for local clock interface output
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
```

```
  1  Sync1 [0/RSP1/CPU0]       0/RSP1/CPU0 T0-SEL-B 1      PRC    1  Locked
     HundredGigE0/5/0/2        0/RSP1/CPU0 T4-SEL-A 1      PRC    1  Available

Selection point: CHASSIS-TOD-SEL (1 inputs, 1 selected)
  Last programmed 6d04h ago, and selection made 6d04h ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: None
    Router scoped : None
  Uses time-of-day selection
  S  Input                     Last Selection Point      Pri  Time  Status
  == ========================  ========================  ===  ====  ===========
  1  Sync1 [0/RSP1/CPU0]       0/RSP1/CPU0 T0-SEL-B 1    100  Yes   Available

Node 0/3/CPU0:
==============
Selection point: ETH_RXMUX (0 inputs, 0 selected)
  Last programmed 9w6d ago, and selection made 9w6d ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: T0-SEL-B T4-SEL-A
    Router scoped : None
  Uses frequency selection

Selection point: LC_TX_SELECT (1 inputs, 1 selected)
  Last programmed 9w6d ago, and selection made 9w6d ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: None
    Router scoped : None
  Uses frequency selection
  Used for local line interface output
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
  24 Sync1 [0/RSP1/CPU0]       0/RSP1/CPU0 T0-SEL-B 1      PRC    1  Available

Node 0/5/CPU0:
==============
Selection point: ETH_RXMUX (1 inputs, 1 selected)
  Last programmed 06:49:27 ago, and selection made 06:49:27 ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: T0-SEL-B T4-SEL-A
    Router scoped : None
  Uses frequency selection
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
  1  HundredGigE0/5/0/2        n/a                         PRC    1  Available

Selection point: LC_TX_SELECT (1 inputs, 1 selected)
  Last programmed 6d04h ago, and selection made 6d04h ago
  Next selection points
    SPA scoped    : None
    Node scoped   : None
    Chassis scoped: None
    Router scoped : None
  Uses frequency selection
  Used for local line interface output
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
```

```
24 Sync1 [0/RSP1/CPU0]       0/RSP1/CPU0 T0-SEL-B 1      PRC    1  Available
```

Configuring Leap Seconds

This procedure describes the steps involved in leap second configuration. The configuration can be executed in two ways:

- By directly providing the **UTC offset value** in the command.
- By providing the path to a **file** in the command, where the UTC offset information is stored (or available).

1. To enter the PTP configuration mode, use **ptp** command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# ptp
```

2. To configure the UTC offset information by providing the offset value directly, use { **utc-offset** {**baseline** | *date* } { *offset-value* } } command in the ptp configuration mode.
   - Using the **baseline** keyword, enter a positive number for the *offset-value* (it is assumed that a negative UTC offset will not be required).
   - **OR** provide a date (in YYYY-MM-DD format) and the *offset-value* . UTC offset used by PTP will be updated on this date. If you do not specify a date, the configuration is applied for the current day, at midnight.

Note In both cases, providing the UTC *offset-value* directly in the command is mandatory.

```
RP/0/RSP0/CPU0:router(config-ptp)# utc-offset baseline 37
```

```
RP/0/RSP0/CPU0:router(config-ptp)# utc-offset 2018-07-01 38
```

3. To configure UTC offset information by providing the path to a file containing the UTC offset information, use { **utc-offset leap-second-file** { *file-path* } } [**poll-frequency** *days*] command in the ptp configuration mode. Optionally, you can provide a polling frequency in days, at which to poll the file for changes. If a frequency for polling is not specified, the file will polled on the day the file is set to expire.

Note The format of this file must be based on the canonical list present at http://www.ietf.org/timezones/data/leap-seconds.list.

```
RP/0/RSP0/CPU0:router(config-ptp)# utc-offset leap-second-file http://<remote-url>
```

```
RP/0/RSP0/CPU0:router(config-ptp)# utc-offset leap-second-file file://<local-path>
poll-frequency 7
```

Verification

To display the current UTC offset value, use **show ptp utc-offset** command.

```
RP/0/RSP0/CPU0:router# show ptp utc-offset

Current offset: +36 seconds (not valid)
Pending leap seconds:
 From 2017-01-01 offset will be +37 seconds
 From 2018-07-01 offset will be +38 second
 From 2019-07-01 offset will be +39 seconds
Source: User-configured
```

To display the current UTC offset value and related details, use **show ptp utc-offset detail** command.

```
RP/0/RSP0/CPU0:router# show ptp utc-offset detail

Current offset: +36 seconds (valid)
Known leap seconds:
From 1996-01-01 offset was +30 seconds
From 1997-07-01 offset was +31 seconds
From 1999-01-01 offset was +32 seconds
From 2006-01-01 offset was +33 seconds
From 2009-01-01 offset was +34 seconds
From 2012-07-01 offset was +35 seconds
From 2015-07-01 offset was +36 seconds
From 2017-01-01 offset will be +37 seconds
Source: file:///test/xxxuser/leapsec/test/list-leap-seconds.list
Expiry date: 2017-12-28
```

Configuring Multiple PTP Profile Interoperability

This procedure describes the steps involved in configuring interoperability for PTP profiles.

1. To configure an interface and then enter the PTP configuration mode, use **interface** and **ptp** commands respectively.

    ```
    RP/0/RSP0/CPU0:router(config)# interface tenGigE 0/0/0/9

    RP/0/RSP0/CPU0:router(config-if)# ptp
    ```

2. To configure PTP profile, use **profile** command in the interface-ptp configuration mode.

    ```
    RP/0/RSP0/CPU0:router(config-if-ptp)# profile interop-slave
    ```

3. To configure interoperability, use **interop** command in the interface-ptp configuration mode.

    ```
    RP/0/RSP0/CPU0:router(config-if-ptp)# interop
    ```

4. To configure the Telecom profile and domain number to interoperate with, use **profile {profile-type}** and **domain** *domain-number* commands in the interface-ptp-interop configuration mode.

    ```
    RP/0/RSP0/CPU0:router(config-if-ptp-interop)# profile g.8275.2

    RP/0/RSP0/CPU0:router(config-if-ptp-interop)# domain 44
    ```

5. To enable conversion of packets on ingress, use **ingress-conversion** command in the interface-ptp-interop configuration mode. The **ingress-conversion** command, converts the packets received from the incoming Announce messages.

```
RP/0/RSP0/CPU0:router(config-if-ptp-interop)#
ingress-conversion
```

6. To explicitly configure the other related parameters, use the respective commands in the interop-ingress submode.

Note Default values are used for parameters that are not explicitly configured during ingress-conversion. For example, default values will be used for parameters like **ClockAccuracy** or **OffsetScaledLogVariance** if they are not explicitly configured.

```
RP/0/RSP0/CPU0:router(config-if-ptp-interop-ingress)#
priority1 10
priority2 10
```

7. To enable conversion of packets on egress, use **egress-conversion** command in the interface-ptp-interop configuration mode. The **egress-conversion** command converts the packets sent through the outgoing Announce messages. The configuration is the same as for ingress conversion.

```
RP/0/RSP0/CPU0:router(config-if-ptp-interop)#
egress-conversion
```

Verification

To display the interop conversions, use **show ptp interop** command.

```
RP/0/RSP0/CPU0:router# show ptp interop tenGigE 0/0/0/9
Egress Conversions:
    Profile:                    Default -> G.8275.2
    Domain:                           0 -> 10
    Priority1:                        1 -> 128
    Priority2:                      100 -> 100
    ClockClass:                      52 -> 140
    ClockAccuracy:                    0 -> 0x21
    OffsetScaledLogVariance:          0 -> 0x4e5d

  Ingress Conversions:
    Profile:                   G.8275.2 -> Default
    Domain:                          10 -> 0
    Master 51.51.51.51:
      Priority1:                      1 -> 100
      Priority2:                      2 -> 254
      ClockClass:                     3 -> 13
      ClockAccuracy:               0x20 -> 0x20
      OffsetScaledLogVariance:   0x4e5d -> 0x4e5d
```

Configuring PTP Telecom Profile Interface

This procedure describes the steps involved to create an interface for PTP ITU-T Telecom Profiles.

Note It is also possible to make these definitions within a global PTP profile and attach them to the interface using the profile command in PTP interface configuration mode.

1. To configure an interface, use **interface** *type interface-path-id* command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1
```

2. To enter the PTP configuration mode for the given interface, use **ptp** command in the interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if)# ptp
```

3. To configure a PTP profile (or specify a previously defined profile), use **profile** *name* command in the ptp-interface configuration mode.

   Note Any additional commands entered in ptp-interface configuration mode overrides the global profile settings.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# profile tele64
```

4. To configure frequency for Sync or Delay-request messages for the given ptp interface, use **sync frequency** *rate* command or **delay-request frequency** *rate* command appropriately in the ptp-interface configuration mode. The valid configurable values are **2, 4, 8, 16, 32, 64 or 128**.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# sync frequency 128
```

```
RP/0/RSP0/CPU0:router(config-if-ptp)# delay-request frequency 128
```

5. To configure duration for different PTP messages, use one of the following commands in the ptp-interface configuration mode: **announce grant-duration** *duration*, **sync grant-duration** *duration*, or **delay-response grant-duration** *duration*. The duration value can be between **60 and 1000 seconds**.

   Note This duration value represents the length of grant that is requested for a port in Slave state and represents the maximum grant-duration allowed when the port is in Master state.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# announce grant-duration 120
```

```
RP/0/RSP0/CPU0:router(config-if-ptp)# sync grant-duration 120
```

```
RP/0/RSP0/CPU0:router(config-if-ptp)# delay-response grant-duration 120
```

6. To configure a timeout value, length of time by when a PTP message must be received (before PTSF-lossSync is raised), use one of the following commands in the ptp-interface configuration mode: **sync timeout** *timeout* or **delay-response timeout** *timeout*. The timeout value can be between **100 to 10000 micro seconds**.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# sync timeout 120
```

```
RP/0/RSP0/CPU0:router(config-if-ptp)# delay-response timeout 120
```

7. To configure a response for unicast-grant invalid-request, use **unicast-grant invalid-request {reduce | deny}** command. The response for requests with unacceptable parameters would either be denied or granted with reduced parameters.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# unicast-grant
invalid-request reduce
```

8. To configure IPv4 or IPv6 address for a PTP master, use **master {ipv4 | ipv6}** *ip-address* command in the ptp-interface configuration mode.

```
RP/0/RSP0/CPU0:router(config-if-ptp)# master ipv4 192.168.2.1
```

```
RP/0/RSP0/CPU0:router(config-if-ptp)# master ipv6 2001:DB8::1
```

9. To override the clock-class received in Announce messages from the specified Master, use **clock-class** *class* command in the ptp-master-interface configuration mode. The class values can range from **0 to 255**.

```
RP/0/RSP0/CPU0:router(config-if-ptp-master)# clock-class 2
```

Verification

To display the PTP interface details, use **show ptp interfaces brief** command.

```
RP/0/RSP0/CPU0:router# show ptp interfaces brief
Fri Feb  9 11:16:45.248 UTC
Intf              Port         Port                    Line
Name              Number       State        Encap      State          Mechanism
--------------------------------------------------------------------------------
BE1               1            Slave        IPv4       up             2-step DRRM
Gi0/0/0/40        2            Master       IPv4       up             2-step DRRM
```

To verify the configured profile details, use **show run interface** *interface-name* command.

```
RP/0/RSP0/CPU0:router# show run interface Gi0/0/0/33

Wed Feb 28 11:49:16.940 UTC
interface GigabitEthernet0/0/0/33
ptp
  profile slave
  multicast target-address ethernet 01-1B-19-00-00-00
  transport ethernet
  port state slave-only
  clock operation two-step
!
ipv4 address 21.1.1.2 255.255.255.0
frequency synchronization
  selection input
  priority 5
  wait-to-restore 0
!
```

Configuring PTP Telecom Profile Clock

This procedure describes the steps involved to configure PTP clock and its settings to be consistent with ITU-T Telecom Profiles for Frequency.

1. To enter the PTP configuration mode, use **ptp** command in the configuration mode.

```
RP/0/RSP0/CPU0:router(config)# ptp
```

2. To enter the PTP-clock configuration mode, use **clock** command in the ptp-configuration mode.

```
RP/0/RSP0/CPU0:router(config-ptp)# clock
```

3. To configure the domain-number for a PTP profile, use **domain** *number* command in the ptp-configuration mode. The allowed domain number range for G.8265.1 profile is between **4 and 23** and the range for G.8275.1 profile is between **24 and 43**.

```
RP/0/RSP0/CPU0:router(config-ptp)# domain 24
```

4. To configure timescale, use **timescale** *source* command in the ptp-clock configuration mode.

```
RP/0/RSP0/CPU0:router(config-ptp-clock)# timescale PTP
```

5. To configure the time-source that will be advertised in Announce messages, use **time-source** *source* command in the ptp-clock configuration mode. The allowed options are: atomic-clock, GPS, hand-set, internal-oscillator, NTP, other, PTP, and terrestrial-radio.

```
RP/0/RSP0/CPU0:router(config-ptp-clock)# time-source GPS
```

6. To exit the ptp-clock configuration mode, use **exit** command.

```
RP/0/RSP0/CPU0:router(config-ptp-clock)# exit
```

7. To configure the desired telecom profile and the clock type for the profile, use **clock profile { g.8265.1 | g.8275.1 | g.8275.2} clock-type {T-GM | T-BC | T-TSC}** command in the ptp configuration mode.

Note The **clock-selection telecom-profile** and **clock-advertisement telecom-profile** commands are deprecated from Release 6.1.2. They are replaced by the **clock profile** command.

```
RP/0/RSP0/CPU0:router(config-ptp)# clock profile g.8275.1 clock-type T-BC
```

Verification

To display the configured PTP clock profile details, use **show run ptp** command.

```
RP/0/RSP0/CPU0:router# show run ptp !
ptp
clock
  domain 24
  profile g.8275.1 clock-type T-BC
!
```

```
profile slave
  sync frequency 16
  announce frequency 8
  delay-request frequency 16
!
profile master
  sync frequency 16
  announce frequency 8
  delay-request frequency 16
!
log
  servo events
  best-master-clock changes
!
!
```

To verify that PTP has been enabled on the router and the device is in LOCKED Phase, use **show ptp platform servo** command.

```
RP/0/RSP0/CPU0:router # show ptp platform servo

Fri Feb  9 11:16:54.568 UTC
Servo status: Running
Servo stat_index: 2
Device status: PHASE_LOCKED
Servo log level: 0
Phase Alignment Accuracy: 1 ns
Sync timestamp updated: 111157
Sync timestamp discarded: 0
Delay timestamp updated: 111157
Delay timestamp discarded: 0
Previous Received Timestamp T1: 1518155252.263409770  T2: 1518155252.263410517  T3:
1518155252.287008362  T4: 1518155252.287009110
Last Received Timestamp T1: 1518155252.325429435  T2: 1518155252.325430194  T3:
1518155252.348938058  T4: 1518155252.348938796
Offset from master:  0 secs, 11 nsecs
Mean path delay   :  0 secs, 748 nsecs
setTime():2  stepTime():1  adjustFreq():10413 adjustFreqTime():0
Last setTime: 1.000000000 flag:1  Last stepTime:-736216, Last adjustFreq:465
```

Configuring Class C Timing Mode

Configuration Example

Use the **timing-accuracy enhanced** command to switch to Class C from Class B mode.

```
Router#configure
Router(config)#frequency synchronization
Router(config-freqsync)#timing-accuracy enhanced
Router(config-freqsync)#commit
```

Verification

Use the **show running-config frequency synchronization** command to verify that you have enhanced to Class C mode from Class B mode.

```
Router#show running-config frequency synchronization
frequency synchronization
 quality itu-t option 1
 clock-interface timing-mode system
 timing-accuracy enhanced
 !
```

System Log

The following system log informs you about the successful mode change from Class B to Class C:

```
RP/0/RSP0/CPU0:Sep 2 15:51:50.784 UTC: dao_tmp[119]:
%PLATFORM-CLKCTRL-5-CLOCK_ACCURACY_ENHANCED : System timing switching to enhanced accuracy
 mode
```

Performance Monitoring for PTP Networks

Table 51: Feature History Table

Feature Name	Release Information	Feature Description
Performance Monitoring	Release 24.3.1	You can now get statistical information with Performance Monitoring in PTP networks, such as clock accuracy, synchronization status, and network delays by defining Performance Monitoring Parameters and Port Specific Parameters. This feature empowers operators with comprehensive performance monitoring and precise time-stamp analysis, offering enhanced granularity for time synchronization in telecommunication networks. By providing detailed insights, it enables operators to make well-informed decisions and take proactive actions to ensure optimal network performance. The feature introduces these changes: **CLI:** • **performance-monitoring** • **show ptp platform performance-counters** • **show ptp dataset performance** **YANG Data Models:** • `Cisco-IOS-XR-ptp-cfg.yang` • `Cisco-IOS-XR-ptp-oper.yang` • `Cisco-IOS-XR-um-ptp-cfg.yang` (see GitHub, YANG Data Models Navigator)

Performance Monitoring in PTP involves tracking and analyzing the performance of PTP networks to ensure accurate time synchronization across devices. This includes monitoring various metrics such as clock accuracy, synchronization status, and network delays. The goal is to identify and address any issues that may affect the precision and reliability of time synchronization in the network.

Performance Monitoring now has the ability to provide performance monitoring and time-stamp analysis information in a PTP network as per Annex J IEEE 1588-2019. This feature also includes additional monitoring granularity for time synchronization in telecommunication networks as per Annex F from the G8275 standard. For more information on PTP, Best TimeTransmitter Clock Algorithm (BTCA), see PTP Overview.

You can use the following parameters to define the Performance Monitoring in a PTP Network:

- Performance Monitoring Parameters
- Port Specific Parameters

Performance Monitoring Parameters

In addition to using the timestamps received from the grandmaster to sync to the grandmaster's clock, the timestamps can now be used to calculate parameters that are of your interest in performance monitoring:

- **TimeTransmitter - TimeReceiver Delay**: corrected propagation time from TimeTransmitter to TimeReceiver.
- **TimeReceiver - TimeTransmitter Delay**: corrected propagation time from TimeReceiver to TimeTransmitter.
- **Mean Path Delay**: mean propagation time over the PTP Communication Path.
- **Offset from TimeTransmitter**: time difference between a TimeTransmitter PTP instance and a TimeReceiver PTP instance as computed by the TimeReceiver PTP instance.

For each of these parameters, you can measure the average, minimum, maximum, and standard deviation for each measurement. These values are calculated and maintained for the following time intervals over the specified time periods:

- **3-minute**: maintained for the current 1-hour period.
- **15-minute**: maintained for the current 24-hour period.
- **1-hour**: maintained for the current 2-hour period.
- **24-hour**: maintained for the current 48-hour period.

The platform actively calculates the end-to-end latency between the TimeTransmitter and TimeReceiver through the Delay-Request-Response-Mechanism (DRRM), allowing Precision Time Protocol (PTP) to seamlessly operate across networks equipped with Transparent clocks, non-PTP aware switches, or a mix of both. Upon a request, PTP dynamically extracts these calculated values from the servo using a platform specific API, allowing you to make proactive changes to the network to ensure precise time synchronization essential for applications that depend on accurate timing.

Additional Port Specific Parameters

The parameters mentioned above apply to the entire Precision Time Protocol (PTP) instance, and there is an additional set of parameters specific to individual ports. These parameters include the counters for various packet types,

- received (rx) and
- transmitted (tx)

It is important to collect and maintain these counters for performance monitoring purposes, which follows the same time intervals and periods as those used for monitoring clock performance.

Port-specific parameters play a crucial role in ensuring accurate time synchronization. These packet types are essential for maintaining the accuracy and reliability of time synchronization in a PTP network:

- **Sync Packets**: These packets are sent by the master clock to the slave clocks to synchronize their time. They contain the precise time at which the packet was sent.
- **Delay Request Packets**: Sent by the slave clock to the master clock, these packets measure the delay between the master and slave clocks. The master clock responds with a Delay Response packet.
- **Follow-Up Packets**: These packets are sent by the master clock immediately after the Sync packet. They contain the exact time the Sync packet was sent, which helps in correcting any delays introduced by the network.
- **Announce Packets**: These packets are used by the master clock to announce its presence and capabilities to the slave clocks. They help in the selection of the best master clock in the network.
- **Management Packets**: These packets are used for configuration and management purposes within the PTP network. They allow for the adjustment of various parameters and settings.

Record format

Record format refers to the structure or layout of data within a record, which is used to store information about time synchronization events and measurements. This format can include various fields such as timestamps, event types, and other relevant data that PTP uses to maintain accurate time synchronization across a network. It is a single buffer for both annexes.

The format is the same for both clock and port performance monitoring parameters that is presented in the operational data. The data is stored over a 48-hour period, resulting in a list of records as per Annex J 1588-2019, composed of the following:

- 1 record for the current 15-minute set of statistics (stored at position 0 in the buffer).
- 96 records for the 15-minute sets of statistics over the last 24-hour period (stored between positions 1-96 in the buffer).
- 1 record for the current 24-hour set of statistics (stored at position 97 in the buffer).
- 1 record for the previous 24-hour set of statistics (stored at position 98 in the buffer).

The data buffer records data at 3-minute intervals over the most recent 1-hour period, creating a list of records that includes:

- 1 record for the current 3-minute set of statistics (stored at position 100 in the buffer).
- 20 records for the 3-minute set of statistics over the last 1-hour period (stored between positions 101-120).
- 1 record for the current 1-hour set of statistics (stored at position 121 in the buffer).
- 1 record for the previous 1-hour set of statistics (stored at position 122 in the buffer).

Configure PTP Performance Monitoring

The purpose of this task is to configure and verify PTP performance monitoring.

Step 1 Configure the **performance-monitoring** command to enable collection of performance monitoring statistics and for the users to make performance monitoring requests.

Example:

```
Router(config)# ptp
Router(config-ptp)# performance-monitoring
Router(config-ptp)# commit
```

Step 2 Run the **sh ptp platform performance-counters** command to display the details of all 123 records.

Example:

The existing command **show ptp platform** is extended to include the performance monitoring data for the local clock.

Note The **detail** mode of the command displays all 123 records, while the **brief** mode displays only the current windows for 15 minutes, 24 hours, 3minutes, and 1hour.

Example:

```
Router#sh ptp platform performance-counters detail

PTP Current record index 15 min: 96
PTP Current record index 3 min: 119

PTP performance monitoring statistics:
====================================================================================================
15 min stats
[0]     12 August 2024 07:08:59 UTC 15 min statistics
----------------------------------------------------------------------------------------------------

                Stat    Min(sec.nsec)        Max(sec.nsec)       Mean(sec.nsec)     Std deviation
         Samples
----------------------------------------------------------------------------------------------------

  Master-slave-delay -000000000.15937      000000000.333       -000000000.1780       000000000.71191
      154
  Slave-master-delay  000000000.319        000000000.16593      000000000.2437       000000000.74103
      154
     mean-path-delay  000000000.322        000000000.334        000000000.327        000000000.4057
       154
  offset-from-master -000000000.16263      000000000.6         -000000000.2108       000000000.72546
      154

----------------------------------------------------------------------------------------------------

           Complete      Valid      PmRef      ServoAtStart     ServoAtEnd         LastServoFlapTime
----------------------------------------------------------------------------------------------------

               FALSE      FALSE       TRUE        PHASE_LOCKED   HOLDOVER              12 Apr 2024
 07:09:09 UTC

==============

….


Router#sh ptp platform performance-counters brief

==============
PTP Current record index 15 min: 96
```

```
PTP Current record index 3 min: 116

PTP performance monitoring statistics:
====================================================================================================
15 min stats
[0]     30 Apr 2024 11:46:07 UTC 15 min statistics
----------------------------------------------------------------------------------------------------

                Stat    Min(sec.nsec)        Max(sec.nsec)       Mean(sec.nsec)     Std deviation
         Samples
----------------------------------------------------------------------------------------------------

  Master-slave-delay  000000000.271        000000000.336        000000000.325       000000000.38386
      13922
  Slave-master-delay  000000000.314        000000000.377        000000000.326       000000000.38526
      13922
     mean-path-delay  000000000.318        000000000.334        000000000.325       000000000.38425
      13922
  offset-from-master -000000000.53         000000000.9         -000000000.0         000000000.369
      13922

----------------------------------------------------------------------------------------------------

           Complete      Valid      PmRef      ServoAtStart     ServoAtEnd         LastServoFlapTime
----------------------------------------------------------------------------------------------------

               FALSE      FALSE       TRUE         FREQ_LOCKED   HOLDOVER               30 Apr 2024
 12:00:33 UTC

==============
```

Step 3 Run the **show ptp dataset performance clock** command to display the performance monitoring data-set details in 15 minutes intervals.

Example:

```
Router#show ptp dataset performance clock

performanceMonitoringDS for the current 15-minute window:
Clock ID ccccfffecccc00, steps removed 1, receiving-port 2:
    Start of time window: Thursday, April 11, 2024 14:18:59
    Measurement is valid
    Period is complete
    Measurement has been taken with reference to system clock
    Master slave delay:
        Average: 50ns
        Min: 50ns
        Max: 70ns
        Std: 1ns
    Slave master delay:
        Average: 51ns
        Min: 51ns
        Max: 71ns
        Std: 2ns
    Mean path delay:
        Average: 52ns
        Min: 52ns
        Max: 72ns
        Std: 3ns
    Offset from master:
        Average: 53ns
        Min: 53ns
        Max: 73ns
        Std: 4ns
```

```
Clock ID aaaabbbeccccc00, steps removed 1, receiving-port 2:
    Start of time window: Thursday, April 11, 2024 14:18:59
    Measurement is not valid
    Period is not complete
    Measurement has been taken with reference to system clock
    Master slave delay:
        Average: 50ns
        Min: 50ns
        Max: 70ns
        Std: 1ns
    Slave master delay:
        Average: 51ns
        Min: 51ns
        Max: 71ns
        Std: 2ns
    Mean path delay:
        Average: 52ns
        Min: 52ns
        Max: 72ns
        Std: 3ns
    Offset from master:
        Average: 53ns
        Min: 53ns
        Max: 73ns
        Std: 4ns
```

Step 4 Run the **show ptp dataset performance port** to display the Performance Monitoring Port Data-set in 15 minutes intervals.

Example:

```
Router#show ptp dataset performance port GigabitEthernet 0/0/0/1

performanceMonitoringPortDS for the current 15-minute window:
Interface GigabitEthernet 0/0/0/1
    Start of time window: Thursday, April 11, 2024 14:18:59
    Measurement is valid
    Period is not complete
    Measurement has been taken with reference to system clock
    Packets                     Sent        Received         Dropped
    ----------------------------------------------------------------
    Announce                       3              83              11
    Sync                           0              32               5
    Follow-Up                      0              31               0
    Delay-Req                     22               0               0
    Delay-Resp                     0              21               7
    Pdelay-Req                     0               7               0
    Pdelay-Resp                    0               0               0
    Pdelay-Resp-Follow-Up          0               0               0
    Signaling                      2               1               0
    Management                     0               0               0
    Other                          0               3              12
                               -----           -----           -----
    TOTAL                         27             178              35
```

Configuring PTP Delay Asymmetry

Table 52: Feature History Table

Feature Name	Release Information	Description
PTP Delay Asymmetry	Release 7.3.1	Any delays on Precision Time Protocol (PTP) paths can impact PTP accuracy and in turn impact clock settings for all devices in a network. This feature allows you to configure the static asymmetry such that the delay is accounted for and the PTP synchronization remains accurate. The **delay-symmetry** command is introduced for this feature.

Configure PTP delay asymmetry to offset the static delays on a PTP path that occur due to different route selection for forward and reverse PTP traffic. Delays can also be due to any node having different delay for ingress or egress path. These delays can impact PTP accuracy due to the asymmetry in PTP. With this feature, you can enable a higher degree of accuracy in the PTP server performance leading to better synchronization between real-time clocks of the devices in a network. Better synchronization between the clocks of different devices in a network leads to a network that performs efficiently.

Configuration of this delay asymmetry provides an option to configure static delays on a client clock for every server clock. You can configure this value in microseconds and nanoseconds. Configured PTP delay asymmetry is also synchronized with the Servo.

Note If you configure multiple PTP delay asymmetries for the same PTP profile, the latest PTP delay asymmetry that you configure is applied to the PTP profile.

A positive value indicates that the server-to-client propagation time is longer than the client-to-server propagation time, and conversely for negative values.

Supported PTP Profiles

The following PTP profiles support the configuration of PTP delay asymmetry:

- PTP over IP (G8275.2 or default profile)
- PTP over L2 (G8275.1)

Restrictions

- PTP delay asymmetry can be configured only on the PTP port of the grandmaster clock, which can either be a boundary clock or an ordinary clock.

- For G875.1 and G875.2 PTP profiles, PTP delay asymmetry is supported for both, client port and dynamic port that act as a client.
- PTP delay asymmetry is supported for fixed cable delay compensation and not for variable delay in the network.
- PTP delay asymmetry can be configured within the range of 3 microseconds and -3 microseconds or 3000 nanoseconds and -3000 nanoseconds.
- Fixed delay can be measured by using any test and measurement tool. Fixed delay can be compensated by using the positive or negative values. For example, if the fixed delay is +10 nanoseconds, configure -10 nanoseconds to compensate the fixed delay.

Configuration

To configure PTP delay asymmetry:

1. Configure an interface with PTP.
2. Configure PTP delay asymmetry on the client side.

Configuration Example

```
/* Configure an interface with PTP. */
Router# configure
Router(config)# interface HundredGigE 0/1/0/0
Router(config-if)# ptp

/* Configure PTP delay asymmetry on the client side. */
Router(config-if-ptp-master)# delay-asymmetry 3 microseconds
Router(config-if-ptp-master)# commit
```

Running Configuration

```
interface preconfigure HundredGigE 0/1/0/0
 ptp
  delay-asymmetry 3 microseconds
```

Verification

To verify if PTP delay asymmetry delay is applied, use the **show ptp foreign-masters** command:

```
Router# show ptp foreign-masters
Sun Nov 1 10:19:21.874 UTC
Interface HundredGigE0/1/0/0 (PTP port number 1)
IPv4, Address 209.165.200.225, Unicast
Configured priority: 1
Configured clock class: None
Configured delay asymmetry: 3 microseconds <------ configured variable delay asymmetry value
Announce granted: every 2 seconds, 300 seconds
Sync granted: 16 per-second, 300 seconds
Delay-resp granted: 16 per-second, 300 seconds
Qualified for 2 minutes, 45 seconds
Clock ID: 80e01dfffe8ab73f
Received clock properties:
Domain: 0, Priority1: 128, Priority2: 128, Class: 6
Accuracy: 0x22, Offset scaled log variance: 0xcd70
Steps-removed: 1, Time source: GPS, Timescale: PTP
Frequency-traceable, Time-traceable
```

```
Current UTC offset: 37 seconds (valid)
Parent properties:
Clock ID: 80e01dfffe8ab73f
Port number: 1
```

To validate the approximate compensated delay value, use the **show ptp platform servo** command:

```
Router# show ptp platform servo
Sat Nov 1 15:17:14.611 UTC
Servo status: Running
Servo stat_index: 2
Device status: PHASE_LOCKED
Servo Mode: Non Hybrid
Servo log level: 0
Phase Alignment Accuracy: -3 ns
Sync timestamp updated: 54754
Sync timestamp discarded: 0
Delay timestamp updated: 55196
Delay timestamp discarded: 0
Previous Received Timestamp T1: 1563984472.036333938 T2: 1563984472.036334935 T3:
1563984472.077066895 T4: 1563984472.077067478
Last Received Timestamp T1: 1563984472.100355188 T2: 1563984472.100356182 T3:
1563984472.139059682 T4: 1563984472.139060266
Offset from master: 0 secs, 6 microsecs <--------------- compensated value, showing 6
microseconds because actual fixed delay is 3 microseconds, we configured 3 microseconds and
 router detected it as fixed delay and tries to compensate.
Mean path delay : 0 secs, 771 nsecs
setTime():1 stepTime():0 adjustFreq():4278
Last setTime: 1563981048.000000000 flag:0 Last stepTime:0 Last adjustFreq:51511
```

Double Failure Clock Class Over-ride

Table 53: Feature History Table

Feature Name	Release Information	Feature Description
PTP Double Failure Clock Class	Release 7.7.1	This feature enables you to configure a clock class that will over-ride the existing class during a state of double-failure where PTP and SyncE are lost. This feature introduces the **double-failure-clock-class** command.

PTP Holdover Traceability suppression

Table 54: Feature History Table

Feature Name	Release Information	Feature Description
PTP Holdover Traceability Suppression	Release 7.3.1	When a device which is configured as a Boundary clock (T-BC) loses synchronization with a quality Primary clock, to ensure that the downstream nodes continue to receive the configured clock class for a specified duration, and it’s traceable you can configure this feature.

When the device loses synchronization with a quality Primary clock, to ensure the downstream nodes continue to receive the configured clock class, and it is traceable you can configure this feature.

This feature enables the device which is configured as a boundary clock (T-BC) with PTP Profiles G.8275.1 or G.8275.2 to send out the configured clock-class as holdover clock-class and the time traceability flag to be set as TRUE for the specified duration. This is to ensure the downstream nodes do not have an impact as this is a deviation from prescribed G.8275.1 ITU-T standards.

Note

- There will be PTP flaps during switchovers or ISSU as the PTP holdover timer is running on the active RSP.
- Once the configured holdover override duration has lapsed and the device is unable to receive quality Primary clock within this duration, the device will send the prescribed default clock class of 165 and the traceability flag will be set as FALSE to advertise loss of clock to downstream nodes.

Configuring PTP Holdover traceability suppression

This section describes how to configure the PTP holdover traceability suppression feature:

```
Router# config
Router(config)# ptp
Router(config-ptp)# holdover-spec-duration 1000
Router(config-ptp)# holdover-spec-clock-class 135
Router(config-ptp)# uncalibrated-traceable-override
Router(config-ptp)# holdover-spec-traceable-override
```

Configuration Examples

Slave Configuration Example

The following example shows a PTP slave configuration:

```
 interface TenGigE 0/1/0/5
 ptp
  profile tp64
  transport ipv4
  port state slave-only
  master ipv4 1.7.1.2
   !
  announce interval 1
   !
 ipv4 address 1.7.1.1 255.255.255.0
!
```

Master Configuration Example

This example shows a PTP master configuration:

```
 ptp
  profile tp64
  transport ipv4
  announce interval 1
 !
 ipv4 address 1.7.1.2 255.255.255.0
 !
```

PTP Hybrid Mode Configuration Example

This example shows the configuration of PTP hybrid mode:

```
ptp
 time-of-day priority 10
 !
interface GigabitEthernet0/1/1/0
 ptp
  transport ipv4
  port state slave-only
  master ipv4 192.168.52.38
   !
  sync frequency 64
  announce interval 1
  delay-request frequency 64
 !
interface GigabitEthernet 0/1/0/1
 ipv4 address 192.168.52.41 255.255.255.0
 speed 100
```

```
 frequency synchronization
  selection input
  priority 10
  wait-to-restore 0
  ssm disable
  time-of-day-priority 100
 !
```

ITU-T Telecom Profiles Configuration Examples

Master global configuration for the telecom profile:

```
-- For G.8265.1 profile --

ptp
 clock
 domain 4
 profile g.8265.1
 !
  profile master
  transport ipv4
  sync frequency 16
  announce interval 1
  delay-request frequency 16
interface gi 0/2/0/4
 ptp
  profile master
  transport ipv4
  clock operation two-step
 !
 ipv4 address 17.1.1.1/24
```

```
-- For G.8275.1 profile --

ptp
 clock
 domain 24
 profile g.8275.1
 !
  profile master
  transport ethernet
  sync frequency 16
  announce interval 1
  delay-request frequency 16
interface gi 0/2/0/4
 ptp
  profile master
  transport ethernet
  multicast target-address ethernet 01-1B-19-00-00-00
  clock operation two-step
 !
 ipv4 address 17.1.1.1/24
```

Slave global configuration for the telecom profile:

```
-- For G.8265.1 profile --
```

```
ptp
 clock
 domain 4
 profile g.8265.1
 !
  profile slave
  transport ipv4
  sync frequency 16
  announce interval 1
  delay-request frequency 16
interface gi 0/1/0/0
 ptp
  profile slave
  transport ipv4
  Master ipv4 18.1.1.1
  port state slave-only
  !
  clock operation two-step
  !
 ipv4 address 18.1.1.2/24

-- For G.8275.1 profile --

ptp
 clock
 domain 24
 profile g.8275.1 clock-type T-TSC
 !
  profile slave
  transport ethernet
  sync frequency 16
  announce interval 1
  delay-request frequency 16
interface gi 0/1/0/0
 ptp
  profile slave
  transport ethernet
  multicast target-address ethernet 01-1B-19-00-00-00
  !
  clock operation two-step
  !
 ipv4 address 18.1.1.2/24



-*- For G.8275.2 profile -*-


ptp
 clock
  domain 44
  profile g.8275.2 clock-type T-TSC
 !
 profile slave
  transport ipv6
  port state slave-only
  sync frequency 64
  announce frequency 8
  unicast-grant invalid-request deny
  delay-request frequency 64
 !
 log
  servo events
  best-master-clock changes
```

```
 !
!
interface GigabitEthernet0/2/0/12
 ptp
  profile slave
  master ipv6 30::2
  !
 !
 ipv6 address 30::1/64
!
```

Global configuration with clock type as T-Boundary Clock (**T-BC**) for the telecom profile:

```
-- For G.8275.1 profile --

ptp
 clock
 domain 24
 profile g.8275.1 clock-type T-BC
 !
  profile master
  transport ethernet
  sync frequency 16
  announce interval 1
  delay-request frequency 16
  exit
  profile slave
  transport ethernet
  sync frequency 16
  announce interval 1
  delay-request frequency 16
  exit
interface gi 0/2/0/4
 ptp
  profile slave
  transport ethernet
  multicast target-address ethernet 01-1B-19-00-00-00
  !
  clock operation two-step
  !
 ipv4 address 17.1.1.2/24
interface gi 0/2/0/0
 ptp
  profile master
  transport ethernet
  multicast target-address ethernet 01-1B-19-00-00-00
  clock operation two-step
  !
 ipv4 address 18.1.1.1/24
```

Note When G.8275.1 profile is configured on a 100G interface, keywords **commit replace** and **rollback config last 1** does not work and the router configuration rollback fails entirely. Use **rollback config last 1 best-effort** instead.

```
-*- For G.8275.2 profile -*-
ptp
```

```
 clock
  domain 44
  profile g.8275.2 clock-type T-BC
 !
 profile slave
  transport ipv6
  port state slave-only
  sync frequency 64
  announce frequency 8
  unicast-grant invalid-request deny
  delay-request frequency 64
 !
 profile master
  transport ipv6
  sync frequency 64
  announce frequency 8
  unicast-grant invalid-request deny
  delay-request frequency 64
 !
 log
  servo events
  best-master-clock changes
 !
!


interface GigabitEthernet0/2/0/11
 ptp
  profile master
 !
 ipv6 address 30::1/64
!

interface GigabitEthernet0/2/0/12
 ptp
  profile slave
  master ipv6 40::2
  !
 !
 ipv6 address 40::1/64
!
```

CHAPTER 16

Network Synchronization Design Best Practices

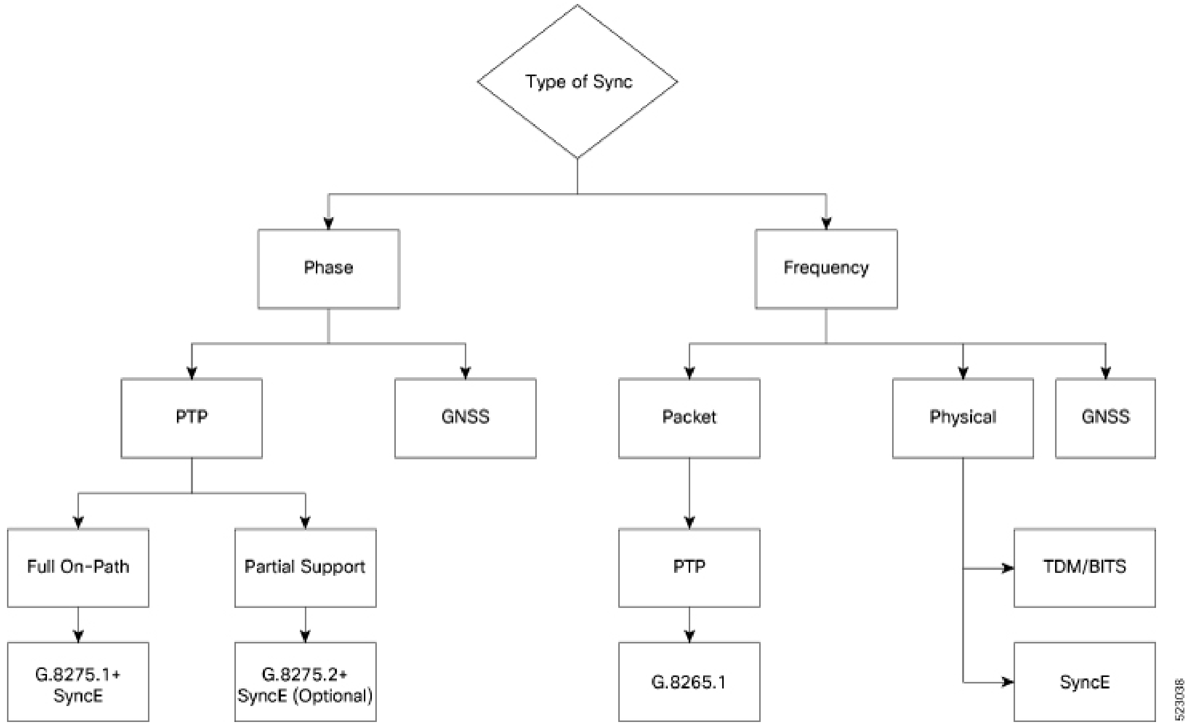
This chapter provides guidelines and best practices to follow when designing timing requirements for your network.

Network Synchronization Design Best Practices

The synchronization of a network is essential for ensuring that all devices in a network run on the same clock time. It also ensures that the applications in the network function correctly. To design your network synchronization accurately, you must have a clear understanding of your network requirements, timing budget, application requirements, and the desired level of synchronization accuracy. This section describes some best practices to follow when designing your network synchronization.

Network Synchronization Decision Tree

Use the network synchronization decision tree for determining the appropriate synchronization solution for your network deployment. Network synchronization helps in ensuring that the network operates with accurate and synchronized time.



General Guidelines for Successful Synchronization Deployments

Network synchronization is crucial for maintaining reliable and efficient network operations, ensuring data integrity, complying with regulations, and facilitating troubleshooting and management tasks. The following guidelines help in deploying successful network synchronization for your network:

- Ensure that you use a standards-based solution designed for your need. For example, use the correct profile.
- Configure the appropriate clock source for your network. It can be Global Navigation Satellite System (GNSS) based such as a Global Positioning System (GPS) clock, or a Precision Time Protocol (PTP) grandmaster clock.
    - Frequency synchronization requires Building Integrated Timing Supply (BITS) or synchronous Ethernet, and Phase synchronization requires PTP and/or GNSS.
    - Use a combination of GNSS over the air and/or PTP or synchronous Ethernet over transport.

  For more information on Configuring Frequency Synchronization and Configuring Precision Time Protocol, refer to *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.
- Set up the synchronization protocols that are required, which includes PTP, Network Time Protocol (NTP), or synchronous Ethernet.
    - NTP uses the system clock for logging events in the system, or to show clock output, whereas PTP and GNSS work on the IEEE 1588 hardware clock in the system.
    - The NTP clock of a node can't be used to synchronize the downstream network using PTP. However, a node can synchronize its NTP clock with the available PTP or GNSS clock.

> **Note** Most NTP implementations are software-based. Software-based time synchronization is less accurate than hardware-based synchronization, but it’s still useful for applications where low levels of accuracy, such as 10's or 100's of milliseconds, are acceptable.

    - Use PTP for phase synchronization in the absence of a GNSS.
    - Synchronous Ethernet (SyncE) is a recommendation from ITU Telecommunication Standardization Sector (ITU-T) on how to deliver a frequency in a network. If you require a frequency-only synchronization solution, use SyncE instead of PTP.

- Configure the appropriate synchronization profiles and preferences for your network. It might include the accuracy, priority, and other parameters that determine how your network handles synchronization events.
- Design your network for phase synchronization with optimal time error budgets.
    - Use boundary clocks to reduce time error and to reset Packet Delay Variation (PDV).
    - Ensure that PTP awareness is implemented consistently throughout, including the transport system, and that boundary clocks accurately transmit time to minimize accumulated time error.
- For phase synchronization, use a hybrid clock that incorporates both SyncE and PTP.

  For more information on Configuring PTP Hybrid Mode, refer to *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.
- Reduce the number of hops:
    - Distribute sources of time to meet the budget. If you have too many hops, install a GNSS receiver further out into the network.
    - Don’t centralize two Primary Reference Time Clocks (PRTC) and Telecom Grandmasters (T-GM) in two different locations and try to run a synchronization signal accurately across the whole network.
- Minimize Packet Delay Variation (PDV) and jitter. Ensure that microwaves, Gigabit-capable Passive Optical Networks (GPON), Digital Subscriber Line (DSL), and Dense Wavelength Division Multiplexing (DWDM) are PTP aware.
- Monitor your synchronization deployment to ensure that it’s functioning correctly and meeting your desired level of accuracy.

  For more information, refer to Verifying the Frequency Synchronization Configuration in the *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.
- Be aware of any relevant industry standards and practices when deploying synchronization.

Guidelines for Phase Synchronization Deployments

Follow these guidelines for phase synchronization deployments.

- Set up the necessary network infrastructure to support phase synchronization. It includes installing timing devices such as GPS receivers, synchronous Ethernet interfaces, and timing servers.

- Configure the phase synchronization protocols such as setting up PTP as appropriate.
- As best practice, use the G.8275.1 telecommunication profile standard with complete on-path support, including Layer-2 multicast in combination with SyncE.
- Minimize phase time error by performing the following tasks:
  - Remove asymmetric routing issues.
  - Reduce the number of hops, unless telecommunication grandmaster (T-GM) clocks are deployed in the preaggregation network.
  - Decrease PDV or packet jitter.
- If you use IP protocols for PTP, you can run into issues with rerouting, asymmetric routing, Equal Cost Multi-Path (ECMP), bundles, and so on.
- If you need tight timing budgets over many hops, ensure that your hardware supports the highest levels of clock accuracy.
- For GNSS deployments:
  - Meet all the requirements for cable and antenna installations.
  - Consult with a professional if you don't have experience with GNSS installation and calibration.
- Make sure that your deployment is working as intended. Monitor it regularly to identify any potential issues.
- Consult with Cisco technical support if you encounter any issues or have questions.

Note When PTP is used with MACsec, achieving high accuracy can be challenging. PTP requires exact timestamping to maintain tight network synchronization. MACsec affixes and detaches a header that is between 24–32 bytes in size. This process can lead to significant inconsistencies in the time delays between where the link is connected and the location where the egress timestamps are applied.

PTP over IP Network Design

When using networks to carry frequency over Precision Time Protocol over Internet Protocol (PTPoIP), the goal is to minimize Packet Delay Variation (PDV) by reducing the number of hops. Use the following guidelines:

- The placement of the telecom grandmaster (T-GM) clock plays an important role in ensuring that the network operates within your timing budget. For example, place a pair of T-GM clocks in a centralized location only if the network has a small number of hops. In larger networks with multiple hops, it may be necessary to distribute T-GM clocks throughout the network to ensure proper timing management at each hop.
- Use a dedicated frequency synchronization protocol such as synchronous Ethernet or 1588v2, which is designed specifically to maintain precise frequency synchronization between devices.

- Use the G.8265.1 standard. Frequency synchronization using the G.8265.1 standard is a way to make sure multiple devices on a network are operating at the same frequency, allowing for more accurate and reliable communication.
- Configure Quality of Service (QoS) policies to prioritize network traffic and reduce delays. This can be done by using traffic shaping, traffic policing, and queue management.

Selecting the Correct Profile For Network Synchronization

G.8275.1 PTPoE

G.8275.1 is a technical specification standard for Precision Time Protocol over Ethernet (PTPoE). It defines how you can use the Precision Time Protocol (PTP) to synchronize clocks over Ethernet networks with layer 2 multicast. PTPoE is an extension of PTP that allows it to be used over Ethernet networks. It's used in applications where precise time synchronization is required.

For more information, refer to G.8275.1 Profile in the *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.

G.8275.2 PTPoIP

G.8275.2 is a technical specification standard for Precision Time Protocol over Internet Protocol (PTPoIP). It defines the use of the Precision Time Protocol (PTP) over packet-based networks such as Internet Protocol (IP) networks, to provide precise time synchronization of network devices.

For more information, refer to G.8275.2 in the *System Management Configuration Guide for Cisco ASR 9000 Series Routers*.

Feature Adaptability on Each Profile

The following table lists the adaptability of features on each profile:

Feature	G.8275.1 PTPoE	G.8275.2 PTPoIP
Network Model	Full on-path support	Partial on-path support
IP Routing	Not applicable	Can cause issues in rings and asymmetry from a number of causes
Transit Traffic	Not allowed	Can result in jitter and asymmetry
Performance	Optimal	Variable
Configuration Model	Physical port	L3 device
PTP over Bundles	No issues	Work in progress for Telecom Boundary Clocks (T-BC)
Asymmetry	Reduced due to T-BC on every node	Optimal when deployed as a Partial Support Telecom Boundary Clock (T-BC-P)

Feature	G.8275.1 PTPoE	G.8275.2 PTPoIP
PDV/Jitter	Reduced due to T-BC on every node	Optimal when deployed as a T-BC-P

Reducing Asymmetry

Asymmetry occurs in a PTP unaware network for the following scenarios:

- When routing large networks, complex topologies, rings, and Equal-cost multi-path (ECMP)
- When using PTP unaware transit nodes, especially with varying traffic patterns
- In the transport layer such as Passive Optical Network (PON), cable, DWDM, and complex optics

Note Every 2 seconds of asymmetry results in 1 microsecond of time error.

To reduce asymmetry in a PTP unaware network:

- Use QoS: QoS can help reduce asymmetry in an unaware network.
- Implement Telecom Boundary Clocks (T-BC): T-BCs can handle asymmetry in the nodes when implemented correctly.

Reducing Packet Delay Variation

To reduce the effects of Packet Delay Variation (PDV) on PTP clock recovery, you must have a steady layer of packets that arrive in minimum time.

- Implement Telecom Boundary Clocks (T-BC) in the PTP unaware node. T-BC introduces a time reference to the PTP unaware node, which then synchronizes its clock with the T-BC.
- Use a high-quality network connection between the T-BC and the PTP unaware node. A high-quality network connection, such as a dedicated fiber link, can help reduce PDV due to network impairments.

Remediating Transport Asymmetry

Transport asymmetry occurs when data is transported at varying rates in different directions over a communication link, leading to an imbalance in transport. To correct this issue:

- Ensure that your transport layer is PTP aware.

  In optical devices, use a wavelength division multiplexing (WDM) technology such as Optical Service Channel (OSC) for managing your fiber optic infrastructure effectively.

Synchronizing Across Networks

To avoid synchronization issues when connecting to other mobile networks:

- Make sure to align all mobile networks to a common source of time. For example, align mobile networks to the Coordinated Universal Time (UTC) from a Global Navigation Satellite System (GNSS) such as Global Positioning System (GPS).
- Monitor your clocks at the interconnect points.

Note In 5G networks, using standalone GNSS receivers at every radio site may not provide the sub-100 nanosecond accuracy required for the timing requirements of Fronthaul radio systems.