System Management Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.0.x

First Published: 2019-08-01

Last Modified: 2020-03-01

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

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

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
August 2019	Initial release of this document.
March 2020	Republished for Release 7.0.2

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.

- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

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

CHAPTER 1

New and Changed System Management Features

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

System Management Features Added or Modified in IOS XR Release 7.0.x

Feature	Description	Changed in Release	Where Documented
Zero Touch Provisioning	This feature was modified.	Release 7.0.1	Configuring Zero Touch Provisioning, on page 427
Erase Disk Memory from RSPs and Line Cards	The Erase Disk Memory operation clears the disk memory of RSPs and line cards.	Release 7.0.1	Overview of Erase and Wipeout Disk Memory, on page 176
Advance Power Management	This feature enables powering down the unused line card slices.	Release 7.0.1	Advanced Power Management, on page 174
Wipeout Disk Memory from RSPs and Line Cards	This feature deletes data permanently from the disk memory of RSPs and line cards.	Release 7.0.2	Overview of Erase and Wipeout Disk Memory, on page 176

CHAPTER 2

Configuring Profiles

Your router caters to different market segments on the service provider edge space. Your router is capable of supporting a wide range of market segments and features, but to make the software more efficient, you must configure the appropriate profiles to achieve the results you require.

- Different customers have different network architectures, and this puts different scale demands on the router. By configuring the *scale profile*, you can configure your router to accommodate your needs.
- The software supports a wide range of features. To optimize performance, each *feature profile* enables a subset of the total available features for a release. You must configure the appropriate profile to enable the features that you require.

Table 2: Feature History for Configuring Profiles

Release	Modification
Release 3.9.1	The scale profile was introduced
Release 4.0.1	The scale profile configuration was moved to admin mode. The feature profile was introduced.

This model contains the following topics:

Restrictions of Scale Profiles

Video monitoring is not supported with the L3XL scale profile.

Information About Profiles

Information About Scale Profiles

A scale profile is a user-configurable setting that tunes the router to perform more efficiently depending on how the router is being used. You should configure a scale profile before deploying the router to production use.

Your router can be used for different market segments on the service provider edge space. Because different customers have different network architectures, which can place different scale demands on the router, it is important to configure the scale profile so that your router works as efficiently as possible within the architecture that you are using.

Possible scenarios that are taken into account by the scale profile are:

- Use of the router as a Layer 2 transport device, thus requiring the support of high Layer 2 scale numbers.
- Use of the router primarily as a Layer 3 box that provides Layer 3 virtual private network (VPN) services, thus requiring the support of a high number of Layer 3 routes.

There are three scale profiles available on your router:

- The *default scale profile* that supports deployments that require large Layer 2 MAC tables (up to 512,000 entries) and a relatively small number of Layer 3 routes (less than 512,000).
- The *Layer 3 scale profile* that supports deployments that require more Layer 3 routes (up to 1 million) and smaller Layer 2 MAC tables (less than 128,000 entries).
- The *Layer 3 XL scale profile* that supports deployments that require a very large number of Layer 3 routes (up to 1.3 million) and minimal Layer 2 functionality. Note that the support for up to 1.3 million routes is split into IPv4 scaled support and IPv4/IPv6 scaled support. You can configure up to 1.3 million IPv4 routes, or up to 1 million IPv4 routes with 128,000 IPv6 routes. The layer 3 XL scale profile does not support video monitoring.

  You can increase the memory available for BGP by configuring the Layer 3 XL profile on the Cisco ASR9000 Series Router using the **hw-module profile scale l3xl** command. However, this reduces the memory available for some other processes. To activate the new profile, you need to manually reboot the system.

  The memory for BGP and the other processes can be verified by using the following commands before and after the configuration:

  - **show processes memory detail**
  - **show bgp process performance-statistics | include RLIMIT** : This command is available only from Cisco IOS-XR release 6.1.x onwards.

Information About Feature Profiles

To allow sufficient computation capabilities within the router, the available features within the Cisco IOS XR software image are bundled. A feature profile determines which bundle of features is available for you to use.

Although you can always configure a feature, if the feature is not supported by the active feature profile, you cannot use it.

There are two feature profiles available on your router:

- The *default profile* that supports all Cisco IOS XR software features except for IEEE 802.1ah provider backbone bridge (PBB).
- The *Layer 2 profile* that supports all Cisco IOS XR software features including IEEE 802.1ah PBB, but does not support IPv6, reverse-path forwarding (RPF) or netflow.

If the feature profile that you have configured on your router does not support a feature that you have configured, warning messages are displayed on the console, and the feature does not work. A configured feature profile takes affect only after you reload all the line cards on the router.

Relationship Between Scale and Feature Profiles

Although you are not limited in your selection of scale and feature profiles in relation to each other, Cisco recommends using the scale and feature profiles together as indicated here:

Table 3: Interaction between Scale and Feature Profiles

	Default Feature Profile	**Layer 2 Feature Profile**
Default Scale Profile	Up to 512 K Layer 3 CEF[1] scale	PPB[2]
Layer 3 Scale Profile	Up to 1.0 M Layer 3 CEF scale Less than 128 K MAC entries	Not recommended
Layer 3 XL Scale Profile	Up to 1.3 M Layer 3 CEF scale	Not recommended

1 Cisco Express Forwarding
2 provider backbone bridge

Other pairs are not recommended. Note that the Layer 3 XL scale profile does not support video monitoring.

Configure iTCAM profile

Both A99-12X100GE and A9K-4X100GE line cards have an internal TCAM of 5MB. You can recarve internal TCAM partition at a Global Configuration level to increase entries on the L2 table and V6 table. Recarving of the TCAM partition helps in the optimal and efficient utilisation of the available memory.

Table 4: Recarving iTCAM profile

Default Limits	**Recarving Limits**
1K entries for L2	4K entries for L2 table
24K limit for V4 entries	Adjusted as required for V4 table
1.75K limit for V6 entries	3.25L entries for V6 table

Restrictions

- This configuration is supported only on A99-12X100GE and A9K-4X100GE line cards.
- For 32-bit IOS-XR, perform this configuration in the Admin Configuration mode.
- For 64-bit IOS-XR perform this configuration in the Global Configuration mode.
- Unless you reload the line cards after the configuration of iTCAM profile on the linecards, the configuration does not take effect.

Configuration Example

To configure iTCAM profile of linecards, use the following steps:

1. Enter the Administration Configuration mode.
2. Configure iTCAM profile of line cards as **to-profile-se1** to recarve TCAM partition of line cards and change the entries to accommodate more L2 or V6 entries in the L2 table and V6 table.

> **Note** If you configure the iTCAM profile as **to-default**, it enables default TCAM entries present in the linecards.

3. Reload the A99-12X100GE and A9K-4X100GE line cards line cards in the chassis.

Configuration

```
/* Enter the Administration Configuration mode and configure iTCAM profile on an interface
 for line cards as to-profile-se1 or to-default. */
Router(admin-config)# hw-module profile itcam to-profile-se1 location 0/0/CPU0
Sun Mar  3 07:44:23.066 UTC
In order to activate this new internal tcam partition profile, you must manually reload the
 line card.
Router(admin-config)# commit

/* Reload the entire router or all the line cards in the chassis. */
```

Verification

To verify the increase in the limits of L2 and V6 entries in the L2 table and V6 tabl for line cards on an interface, use the **show prm server tcam summary all all detail all location *location*** command. In the output, you can see that L2 entries have increased to 4K in the L2 table, V4 entries have reduced to 1.5K in the V4 table, and V6 entries have increased to 3.5K in the V6 table.

```
 Router# show prm server tcam summary all all detail np3 location 0/0/CPU0
Wed Mar 13 21:37:43.743 UTC

                Node: 0/0/CPU0:
----------------------------------------------------------------

TCAM summary for NP3:

  TCAM Logical Table: TCAM_LT_L2 (1)
    Partition ID: 0, valid entries: 2, free entries: 22
    Partition ID: 1, valid entries: 0, free entries: 24
    Partition ID: 2, valid entries: 0, free entries: 24
    Partition ID: 3, valid entries: 0, free entries: 2012.
```

```
   Partition ID: 4, valid entries: 2, free entries: 2010
 TCAM Logical Table: TCAM_LT_ODS2 (2), max entries: 15360, num free: 15237
   Application ID: NP_APP_ID_IFIB (0).
     VMR ID:    1, used entries:   45, allocated entries:  123
    Total vmr_ids per app id: 1, Total used entries per app id: 45 Total allocated entries:
123
   Application ID: NP_APP_ID_QOS (1)
    Total vmr_ids per app id: 0, Total used entries per app id: 0 Total allocated entries:
0
   Application ID: NP_APP_ID_ACL (2)
    Total vmr_ids per app id: 0, Total used entries per app id: 0 Total allocated entries:
0
   Application ID: NP_APP_ID_AFMON (3)
    Total vmr_ids per app id: 0, Total used entries per app id: 0 Total allocated entries:
0
   Application ID: NP_APP_ID_LI (4)
     VMR ID:    2, used entries:    0, allocated entries:    0
    Total vmr_ids per app id: 1, Total used entries per app id: 0 Total allocated entries:
0
   Application ID: NP_APP_ID_PBR (5)
    Total vmr_ids per app id: 0, Total used entries per app id: 0 Total allocated entries:
0
 TCAM Logical Table: TCAM_LT_ODS8 (3), max entries: 3328, num free: 3295
   Application ID: NP_APP_ID_IFIB (0).
     VMR ID:    1, used entries:   33, allocated entries:   33
    Total vmr_ids per app id: 1, Total used entries per app id: 33 Total allocated entries:
33
   Application ID: NP_APP_ID_QOS (1)
    Total vmr_ids per app id: 0, Total used entries per app id: 0 Total allocated entries:
0
   Application ID: NP_APP_ID_ACL (2)
    Total vmr_ids per app id: 0, Total used entries per app id: 0 Total allocated entries:
0
   Application ID: NP_APP_ID_PBR (5)
    Total vmr_ids per app id: 0, Total used entries per app id: 0 Total allocated entries:
0
   Application ID: NP_APP_ID_EDPL (6)
    Total vmr_ids per app id: 0, Total used entries per app id: 0 Total allocated entries:
0

RP/0/RSP1/CPU0:VKG6#
```

How to Configure Profiles

Configuring the Scale Profile

Before you deploy your router, you should configure the scale profile to make the system most efficient for your specific network architecture.

Before you begin

In general, the route switch processor (RSP) with 6 GB of memory is capable of loading 1.3 million IPv4 routes. For large scale routes like 4 million, 12 GB of memory is required.

The RSP440 supports 1.3 million IPv4 routes with the default memory.

> **Note** The scale profile should be configured in the administration configuration. If you previously configured the L3 scale profile in the global configuration, the following limitations apply:
>
> - If the scale profile is set only in the global configuration, the setting takes affect.
> - Scale profile settings in the administration configuration override scale profile settings in the global configuration.
> - Cisco recommends that you configure all scale profile settings in the administration configuration and remove the global configuration settings. For more information, refer to Removing the Scale Profile from the Global Configuration, on page 11.

SUMMARY STEPS

1. **admin**
2. **configure**
3. **hw-module profile scale** {**default** | **l3** | **l3xl**}
4. Use the **commit** or **end** command.
5. **reload location all**
6. **show running-config**
7. **show hw-module profile**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**configure** **Example:** `RP/0/RSP0/CPU0:router(admin)# configure`	Enters administration configuration mode.

Step 3	**hw-module profile scale** {**default** | **l3** | **l3xl**}
	Example:
	`RP/0/RSP0/CPU0:router(admin-config)# hw-module profile`
	`scale l3xl`
	
	`Sun Nov 14 10:04:27.109 PST`
	`In order to activate this new memory resource profile,`
	`you must manually reboot the system.`
		Specifies the scale profile for the router.
		• **default**—efficient for deployments that require large Layer 2 MAC tables (up to 512,000 entries) and a relatively small number of Layer 3 routes (less than 512,000).
		• **l3**—efficient for deployments that require more Layer 3 routes (up to 1 million) and smaller Layer 2 MAC tables (less than 128,000 entries).
		• **l3xl**—efficient for deployments that require a very large number of Layer 3 routes (up to 1.3 million) and minimal Layer 2 functionality. Note that the support

	Command or Action	Purpose
		for up to 1.3 million routes is split into IPv4 scaled support and IPv4/IPV6 scaled support. You can configure up to 1.3 million IPv4 routes, or up to 1 million IPv4 routes with 128,000 IPv6 routes.
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	**reload location all** **Example:** `RP/0/RSP0/CPU0:router(admin)# reload location all`	Reloads the entire router or all line cards in the chassis. If you are changing the scale profile to, or from, one of the Layer 3 scale profile values, you must perform a reload of the entire system before the change is enabled.

Step 6	**show running-config**
	Example:
	`RP/0/RSP0/CPU0:router(admin)# show running-config`
	`hw-module profile scale`
		Displays the configured scale profile.

Step 7	**show hw-module profile** **Example:** `RP/0/RSP0/CPU0:router# show hw-module profile scale`	Displays the active scale profile. If the scale profile is different than the configured profile, the line cards have not been reloaded as required for the scale profile configuration to take place.

Configuring the Feature Profile

Before deploying your router you should determine that the feature profile is consistent with the features that you need to use. If it is not, use this task to configure a different profile.

SUMMARY STEPS

1. **admin**
2. **configure**
3. **hw-module profile feature**{**default** | **l2**}
4. Use the **commit** or **end** command.
5. **reload location** {**all** | *node-id*}
6. **show running-config**
7. **show hw-module profile feature**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**configure** **Example:** `RP/0/RSP0/CPU0:router(admin)# configure`	Enters administration configuration mode.

Step 3	**hw-module profile feature**{**default** | **l2**}
	Example:
	`RP/0/RSP0/CPU0:router(admin-config)# hw-module profile feature l2`
	
	`Wed Dec  8 08:29:54.053 PST`
	`L2 feature profile does NOT support the following features:`
	`IPv6, RPF, Netflow.`
	`In order to activate this new memory resource profile,`
	`you must manually reboot the line cards.`
		Specifies the feature profile for the router.
		• **default**—supports all features except provider backbone bridge (PBB).
		• **l2**—supports PBB, but does not support IPv6, reverse-path forwarding (RPF) and netflow.

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	**reload location** {**all** | *node-id*} **Example:** `RP/0/RSP0/CPU0:router(admin)# reload location 0/0/cpu0`	Reloads a line card. Before the feature profile configuration becomes effective, you must reload all line cards in the router. Use the **reload location** *node-id* command for each line card separately.

Step 6	**show running-config**
	Example:
	`RP/0/RSP0/CPU0:router(admin)# show running-config`
	
	`hw-module profile feature`
		Displays the configured feature profile.

	Command or Action	Purpose
Step 7	**show hw-module profile feature** **Example:** `RP/0/RSP0/CPU0:router# show hw-module profile feature all`	Displays the active feature profile. If the active profile is different from the configured profile, the line cards have not been reloaded as required for the feature profile configuration to take place.

What to do next

If you see warning messages to the console indicating that the active feature profile does not match the configured profile, you must reload the affected line card so that the configured profile matches the active profile.

```
LC/0/1/CPU0:Nov 5 02:50:42.732 : prm_server[236]: Configured 'hw-module
profile feature l2' does not match active 'hw-module profile feature default'.
You must reload this line card in order to activate the configured profile on
this card or you must change the configured profile.
```

If you see warning messages to the console indicating that some features do not match the feature profile, you should either change the feature profile configuration, or remove the non-supported features.

```
LC/0/1/CPU0:Nov 5 02:50:42.732 : prm_server[236]: Active 'hw-module profile
feature l2' does not support IPv6, RPF, or Netflow features. Please remove all
unsupported feature configurations.

*"hw-module profile feature" syntax applies to Trident and Lightspeed based line cards;
therefore the limitations of IPv6, reverse-path forwarding (RPF) and Netflow do
not apply to either Tomahawk or Typhoon based line cards.
```

Removing the Scale Profile from the Global Configuration

If a scale profile is configured in the global configuration, you should duplicate the configuration in the administration configuration, and remove the global configuration as described here.

Note If you do not move the scale profile setting to the administration configuration, the configuration in global configuration mode takes affect.

If the scale profile is configured in both the global configuration and administration configuration, the setting in the administration configuration takes precedence.

SUMMARY STEPS

1. **show running-config | file** *new-config-file*
2. Remove the line with the command **hw-module profile scale** from the file created in the previous step.
3. **configure**
4. **load** *new-config-file*
5. **commit replace**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show running-config | file** *new-config-file* **Example:** `RP/0/RSP0/CPU0:router# show running-config | file new-config-file`	Copies the contents of the running configuration to a file.
Step 2	Remove the line with the command **hw-module profile scale** from the file created in the previous step.	Takes out the profile command that is configured in the global configuration.
Step 3	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 4	**load** *new-config-file* **Example:** `RP/0/RSP0/CPU0:router(config)# load new-config-file`	Replaces the running configuration with the edited file.
Step 5	**commit replace** **Example:** `RP/0/RSP0/CPU0:router(config)# commit replace`	Commits the changed configuration to the router.

Additional References

Related Documents

Related Topic	Document Title
Profile commands	*Hardware Redundancy and Node Administration on the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*
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 and RFCs

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

MIBs

MIB	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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 3

Configuring Manageability

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

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

Table 5: Feature History for Configuring Manageability on Cisco IOS XR Software

Release 3.7.2	This feature was introduced
Release 3.9.0	The ability to enable XML requests over Secure Socket Layer (SSL) was introduced. The ability to configure an idle timeout for the XML agent was introduced.
Release 4.0.0	The ability to configure a dedicated agent to receive and send messages via a specified VPN routing and forwarding (VRF) instance was introduced. The ability to control CPU time used by the XML agent was introduced.

This module contains the following topics:

Information About XML Manageability

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

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

How to Configure Manageability

Configuring the XML Agent

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**xml agent** [**ssl**] **Example:** `RP/0/RSP0/CPU0:router:router(config)# xml agent`	Enables Extensible Markup Language (XML) requests over a dedicated TCP connection and enters XML agent configuration mode. Use the **ssl** keyword to enable XML requests over Secure Socket Layer (SSL).
Step 2	**iteration on size** *iteration-size* **Example:** `RP/0/RSP0/CPU0:router:router(config-xml-agent)# iteration on size 500`	Configures the iteration size for large XML agent responses in KBytes. The default is 48.
Step 3	**session timeout** *timeout* **Example:** `RP/0/RSP0/CPU0:router:router(config-xml-agent)# session timeout 5`	Configures an idle timeout for the XML agent in minutes. By default, there is no timeout.
Step 4	**throttle** {**memory** *size* | **process-rate** *tags*} **Example:** `RP/0/RSP0/CPU0:router:router(config-xml-agent)# throttle memory 300`	Configures the XML agent processing capabilities. • Specify the throttle memory size in Mbytes per session. Values can range from 100 to 600. In IOS XR 64 bit, the values range from 100 to 1024. The default is 300. • Specify the process-rate as the number of tags that the XML agent can process per second. Values can range from 1000 to 30000. By default the process rate is not throttled.
Step 5	**vrf** { **default** | *vrf-name*} [**access-list** *access-list-name*] **Example:**	Configures the dedicated agent or SSL agent to receive and send messages via the specified VPN routing and forwarding (VRF) instance.

	Command or Action	Purpose
	RP/0/RSP0/CPU0:router:router(config-xml-agent)# vrf my-vrf	

Configuration Examples for Manageability

Enabling VRF on an XML Agent: Examples

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

```
RP/0/RSP0/CPU0:router:router(config)# xml agent
RP/0/RSP0/CPU0:router:router(config-xml-agent)# vrf VRF1
RP/0/RSP0/CPU0:router:router(config-xml-agent)# vrf VRF2
```

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

```
RP/0/RSP0/CPU0:router:router(config)# xml agent
RP/0/RSP0/CPU0:router:router(config-xml-agent)# no vrf VRF2
```

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

```
RP/0/RSP0/CPU0:router:router(config)# xml agent ssl
RP/0/RSP0/CPU0:router:router(config-xml-agent)# vrf VRF1
RP/0/RSP0/CPU0:router:router(config-xml-agent)# vrf VRF2
```

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

```
RP/0/RSP0/CPU0:router:router(config)# xml agent ssl
RP/0/RSP0/CPU0:router:router(config-xml-agent)# no vrf VRF2
```

CHAPTER 4

Configuring Physical and Virtual Terminals

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

> **Note** Before creating or modifying the vty pools, enable the telnet server using the **telnet server** command in Global Configuration mode. See *IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers* and *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers* for more information.

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

For more information about physical and virtual terminals on the Cisco IOS XR software and complete descriptions of the terminal services commands listed in this module, see Related Documents, on page 29. 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 6: Feature History for Implementing Physical and Virtual Templates on Cisco IOS XR Software

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.9.0	No modification.

This module contains the following topics:

Prerequisites for Implementing Physical and Virtual Terminals

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

Information About Implementing Physical and Virtual Terminals

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

Tip You can programmatically manage the physical and virtual terminals using `openconfig-system-terminal.yang` OpenConfig data model. To get started with using data models, see the *Programmability Configuration Guide for Cisco ASR 9000 Series Routers*.

Line Templates

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

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

Line Template Configuration Mode

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

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

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

```
RP/0/RSP0/CPU0:router(config)# line console
RP/0/RSP0/CPU0:router(config-line)#
```

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

- absolute-timeout—Specifies a timeout value for line disconnection.
- escape-character—Changes the line escape character.
- exec-timeout—Specifies the EXEC timeout.
- length—Sets the number of lines displayed on the screen.
- session-limit—Specifies the allowable number of outgoing connections.
- session-timeout—Specifies an interval for closing the connection if there is no input traffic.
- timestamp—Displays the timestamp before each command.
- width—Specifies the width of the display terminal.

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

Line Template Guidelines

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

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

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

The default settings for the default template are described for all commands in line template configuration mode in the *Terminal Services Commands on the Cisco ASR 9000 Series Router* module in *System Management Command Reference for Cisco ASR 9000 Series Routers*.

Note Before creating or modifying the vty pools, enable the telnet server using the **telnet server** command in Global Configuration mode. See *IP Addresses and Services Configuration Guide for Cisco ASR 9000 Series Routers* and *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers* for more information.

Terminal Identification

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

vty Pools

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

- Default vty pool—The default vty pool consists of five vtys (vtys 0 through 4) that each reference the default line template.
- Default fault manager pool—The default fault manager pool consists of six vtys (vtys 100 through 105) that each reference the default line template.

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

When configuring vty pools, follow these guidelines:

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

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

Modifying Templates

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

SUMMARY STEPS

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

DETAILED STEPS

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

Step 2	**line** {**console** | **default**}
	Example:
	`RP/0/RSP0/CPU0:router(config)# line console`
	or
	`RP/0/RSP0/CPU0:router(config)# line default`
		Enters line template configuration mode for the specified line template.
		• **console** —Enters line template configuration mode for the console template.
		• **default** —Enters line template configuration mode for the default line template.

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

Step 4	Use one of the following commands:
	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-line)# end`
	or
	`RP/0/RSP0/CPU0:router(config-line)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		• Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		• Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Creating and Modifying vty Pools

This task explains how to create and modify vty pools.

You can omit Step 3, on page 24 to Step 5, on page 24 if you are configuring the default line template to reference a vty pool.

SUMMARY STEPS

1. **configure**
2. **telnet** {**ipv4** | **ipv6**} **server max-servers** *limit*
3. **line template** *template-name*
4. Configure the terminal attribute settings for the specified line template using the commands in line template configuration mode.
5. **exit**
6. **vty-pool** {**default** | *pool-name* | **eem**} *first-vty last-vty* [**line-template** {**default** | *template-name*}]
7. Use the **commit** or **end** command.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**telnet** {**ipv4** | **ipv6**} **server max-servers** *limit* **Example:** `RP/0/RSP0/CPU0:router(config)# telnet ipv4 server max-servers 10`	Specifies the number of allowable Telnet servers. Up to 100 Telnet servers are allowed. **Note** By default no Telnet servers are allowed. You must configure this command in order to enable the use of Telnet servers.
Step 3	**line template** *template-name* **Example:** `RP/0/RSP0/CPU0:router(config)# line template 1`	Enters line template configuration mode for a user-defined template.
Step 4	Configure the terminal attribute settings for the specified line template using the commands in line template configuration mode.	—
Step 5	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-line)# exit`	Exits line template configuration mode and returns the router to global configuration mode.
Step 6	**vty-pool** {**default** | *pool-name* | **eem**} *first-vty last-vty* [**line-template** {**default** | *template-name*}] **Example:** `RP/0/RSP0/CPU0:router(config)# vty-pool default 0 5 line-template default` or	Creates or modifies vty pools. • If you do not specify a line template with the **line-template** keyword, a vty pool defaults to the default line template. • **default** —Configures the default vty pool. • The default vty pool must start at vty 0 and must contain a minimum of five vtys (vtys 0 through 4).

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config)# vty-pool pool1 5 50 line-template template1`
	or
	`RP/0/RSP0/CPU0:router(config)# vty-pool eem 100 105 line-template template1`
		• You can resize the default vty pool by increasing the range of vtys that compose the default vty pool.
		• *pool-name* —Creates a user-defined vty pool.
		• A user-defined pool must start at least at vty 5, depending on whether the default vty pool has been resized.
		• If the range of vtys for the default vty pool has been resized, use the first range value free from the default line template. For example, if the range of vtys for the default vty pool has been configured to include 10 vtys (vty 0 through 9), the range value for the user-defined vty pool must start with vty 10.
		• **eem** —Configures the embedded event manager pool.
		• The default embedded event manager vty pool must start at vty 100 and must contain a minimum of six vtys (vtys 100 through 105).
		• **line-template** *template-name* —Configures the vty pool to reference a user-defined template.

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

Monitoring Terminals and Terminal Sessions

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

Note The commands can be entered in any order.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose

Step 1	(Optional) **show line** [**aux location** *node-id* | **console location** *node-id* | **vty** *number*]
	Example:
	`RP/0/RSP0/CPU0:router# show line`
		Displays the terminal parameters of terminal lines.
		• Specifying the **show line aux location** *node-id* EXEC command displays the terminal parameters of the auxiliary line.
		• Specifying the **show line console location** *node-id* EXEC command displays the terminal parameters of the console.
		  ◦ For the **location** *node-id* keyword and argument, enter the location of the Route Processor (RP) on which the respective auxiliary or console port resides.
		  ◦ The *node-id* argument is expressed in the format of *rack/slot/module* .
		• Specifying the **show line vty** *number* EXEC command displays the terminal parameters for the specified vty.

Step 2	(Optional) **show terminal** **Example:** `RP/0/RSP0/CPU0:router# show terminal`	Displays the terminal attribute settings for the current terminal line.
Step 3	(Optional) **show users** **Example:** `RP/0/RSP0/CPU0:router# show users`	Displays information about the active lines on the router.

Craft Panel Interface

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

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

Configuration Examples for Implementing Physical and Virtual Terminals

Modifying the Console Template: Example

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

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

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

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

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

```
RP/0/RSP0/CPU0:router# show line console location 0/0/CPU0

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

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

Modifying the Default Template: Example

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Additional References

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

Related Documents

Related Topic	Document Title
Cisco IOS XR terminal services commands	*Terminal Services Commands on the Cisco ASR 9000 Series Router* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*
Cisco IOS XR command master index	*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*
Information about 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*

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 5

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 57. 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 7: 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.

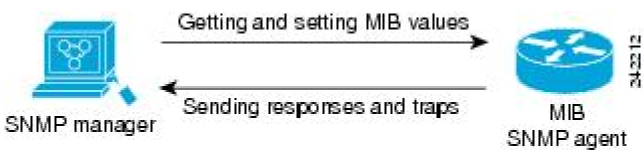
MIB

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 57

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.

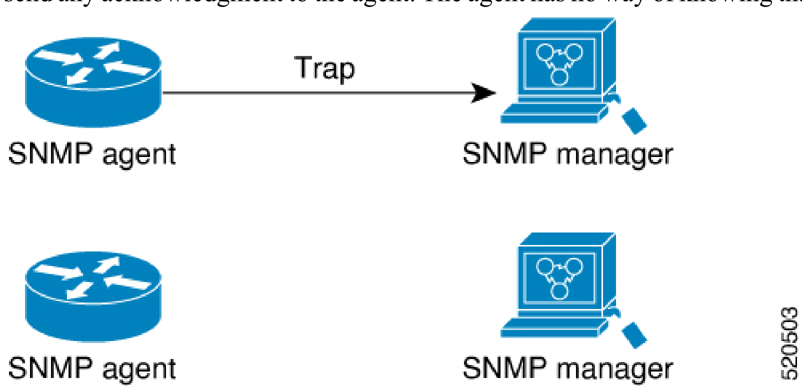
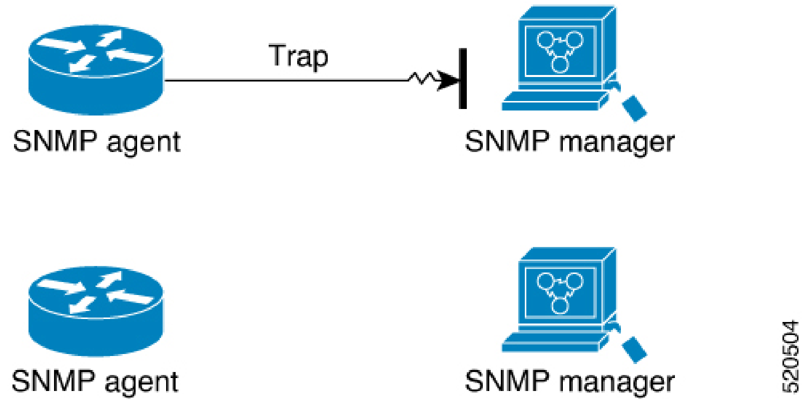


Figure 3: Trap Not Received by the SNMP Manager

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 9: SNMP Security Models and Levels, on page 36 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 8: 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 9: 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[3]-MD5[4] algorithm or the HMAC-SHA[5].
v3	authPriv	HMAC-MD5 or HMAC-SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES[6] 56-bit encryption in addition to authentication based on the CBC[7] 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[8] 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[9] level of encryption.

3 Hash-Based Message Authentication Code
4 Message Digest 5
5 Secure Hash Algorithm
6 Data Encryption Standard
7 Cipher Block Chaining
8 Triple Data Encryption Standard
9 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 10: 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.

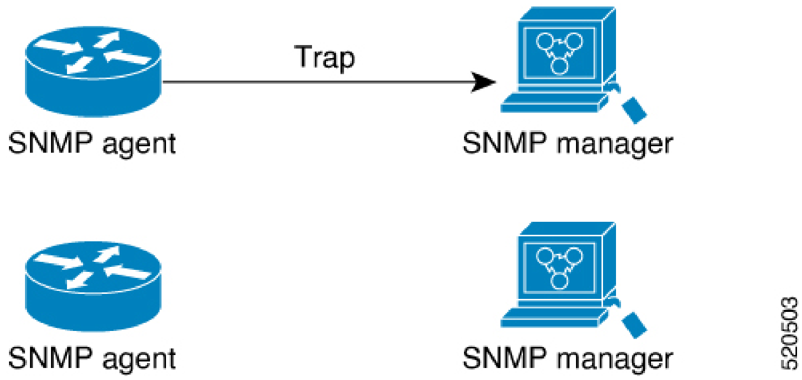
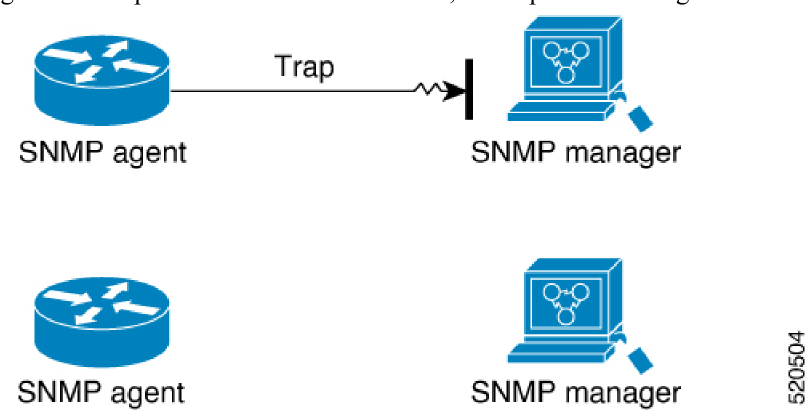


Figure 5: Trap Not Received by the SNMP Manager

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.
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 41 if you have already completed the steps documented under the Configuring SNMPv3, on page 41 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.
Step 4	**snmp-server host** *address* [**traps**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server host 12.26.25.61 traps version 3 noauth userV3noauth`	Specifies SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the notifications.
Step 5	**snmp-server traps** [*notification-type*] **Example:** `RP/0/RP0/CPU0:router(config)# snmp-server traps bgp`	Enables the sending of trap notifications and specifies the type of trap notifications to be sent. • If a trap is not specified with the *notification-type* argument, all supported trap notifications are enabled on the router. To display which trap notifications are available on your router, enter the **snmp-server traps ?** command.

	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.
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:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	(Optional) **snmp-server contact** *system-contact-string* **Example:**	Sets the system contact string.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config)# snmp-server contact Dial System Operator at beeper # 27345`	
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
```

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*

Related Topic	Document Title
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)*

RFCs	Title
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 6

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 75. 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 11: Feature History Table

Feature Name	Release Information	Description
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:

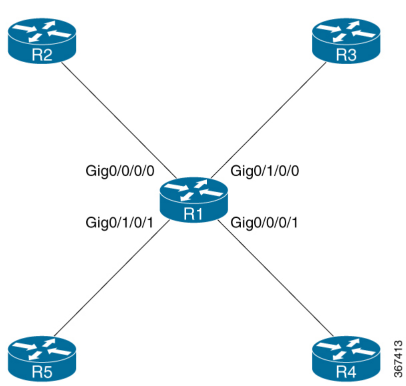
- **Boolean AND function—**When a tracked list has been assigned a Boolean AND function, each object that is defined within a subset must be in an "up" state. This condition enables the tracked object to be in the "up" state.
- **Boolean OR function—**When the tracked list has been assigned a Boolean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



Restrictions for Enhanced Object Tracking

You can perform Enhanced Object Tracking only on physical interfaces and not on virtual interfaces.

The only action you can perform is error-disabling interfaces based on the state of a track (up/down).

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.
Step 3	**type line-protocol state** **Example:**	Creates a track based on the line protocol of an interface.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-track)# type line-protocol state`	
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.
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.

	Command or Action	Purpose
		• 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.
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*	Configures the type of IP route to be tracked, which can consist of either of the following, depending on your router type:

	Command or Action	Purpose

	• **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`
		• *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.
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.
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.

	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.

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:** `RP/0/RSP0/CPU0:router(config)# track track1`	Enters track configuration mode. • *track-name*—Specifies a name for the object to be tracked.
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	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]:`

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-track)# commit`	• 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 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.

	Command or Action	Purpose
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:
	`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`
		Configures object 1, object 2 and object 3 as members of track t1 and with weights 10, 5 and 3 respectively.

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.
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. To enable Enhanced Object, 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:

```
Router# configure
Router(config)# track t1
Router(config-track)# type route reachability route ipv4 192.0.2.1/24
Router(config-track)# action track-down error-disable interface GigabitEthernet0/0/0/1
auto-recover
```

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

```
track track1
 type line-protocol state
  interface GigabitEthernet0/0/0/1
 !
!
track track2
 type line-protocol state
  interface GigabitEthernet0/1/0/1
 !
!
track track3
 type list boolean and
  object track1
  object track2
 !
 action
  track-down error-disable interface GigabitEthernet0/0/0/0 auto-recover
  track-down error-disable interface GigabitEthernet0/1/0/0
```

Verification

To view the status of the track, use the **show track** command:

```
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 GigabitEthernet0/0/0/1 line-protocol
        Line protocol is UP
        7 changes, last change 16:04:28 IST Mon Jul 02 2018
Track track2
        Interface GigabitEthernet0/1/0/1 line-protocol
        Line protocol is UP
        7 changes, last change 16:02:41 IST Mon Jul 02 2018
```

To verify if the interface configured for tracking is disabled, use the **show error-disable** command.

```
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 track, use the **show ipv4 interface brief** command.

```
Router# show ipv4 interface brief
Interface                      IP-Address      Status         Protocol   Vrf-Name
GigabitEthernet0/0/0/0         unassigned       Up              Up       default
GigabitEthernet0/0/0/1         unassigned       Up              Up       default
GigabitEthernet0/1/0/0         unassigned       Up              Up       default
GigabitEthernet0/1/0/1         unassigned       Up              Up       default
```

When the status of track3 is "down", the following output for **show ipv4 interface brief** command is displayed.

```
Router# show ipv4 interface brief
Interface                      IP-Address      Status          Protocol      Vrf-Name
GigabitEthernet0/0/0/0         unassigned      Shutdown        Down          default
GigabitEthernet0/0/0/1         unassigned      Shutdown        Down          default
GigabitEthernet0/1/0/0         unassigned      Shutdown        Down          default
GigabitEthernet0/1/0/1         unassigned      Up              Up            default
```

When track3 goes back to up state, the disabled condition on the interface GigabitEthernet0/0/0/0 is cleared. This condition occurs because **auto-recover** is configured, but interface 0/1/0/0 remains in the disabled state because **auto-recover** is not configured on this interface. The change is reflected in the output of the **show ipv4 interface brief** command.

```
RP/0/0/CPU0:ios#show ipv4 interface brief
Interface                      IP-Address      Status          Protocol   Vrf-Name
GigabitEthernet0/0/0/0         unassigned      Up           Up       default
GigabitEthernet0/0/0/1         unassigned      Up           Up       default
GigabitEthernet0/1/0/0         unassigned      Shutdown     Down     default
GigabitEthernet0/1/0/1         unassigned      Up           Up       default
GigabitEthernet0/1/0/2         unassigned      Shutdown           Down     default
GigabitEthernet0/1/0/3         unassigned      Shutdown           Down     default
```

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 7

Configuring Cisco Discovery Protocol

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

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

For more information about CDP on the Cisco IOS XR software and complete descriptions of the CDP commands listed in this module, refer to Related Documents, on page 85. 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 12: Feature History for Implementing CDP on Cisco IOS XR Software

Release	Modification
Release 3.7.2	This feature was introduced.
Release 3.9.0	No modification.

This module contains the following topics:

Prerequisites for Implementing CDP

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

Information About Implementing CDP

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

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

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

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

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

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

Type-length-value fields (TLVs) are blocks of information embedded in CDP advertisements. Table 13: Type-Length-Value Definitions for CDPv2, on page 78 summarizes the TLV definitions for CDP advertisements.

Table 13: Type-Length-Value Definitions for CDPv2

TLV	Definition
Device-ID TLV	Identifies the device name in the form of a character string.
Address TLV	Contains a list of network addresses of both receiving and sending devices.
Port-ID TLV	Identifies the port on which the CDP packet is sent.
Capabilities TLV	Describes the functional capability for the device in the form of a device type; for example, a switch.
Version TLV	Contains information about the software release version on which the device is running.
Platform TLV	Describes the hardware platform name of the device, for example, Cisco 4500.

TLV	Definition
VTP Management Domain TLV	Advertises the system's configured VTP management domain name-string. Used by network operators to verify VTP domain configuration in adjacent network nodes.
Native VLAN TLV	Indicates, per interface, the assumed VLAN for untagged packets on the interface. CDP learns the native VLAN for an interface. This feature is implemented only for interfaces that support the IEEE 802.1Q protocol.
Full/Half Duplex TLV	Indicates status (duplex configuration) of CDP broadcast interface. Used by network operators to diagnose connectivity problems between adjacent network elements.

How to Implement CDP on Cisco IOS XR Software

Enabling CDP

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**cdp** **Example:** `RP/0/RSP0/CPU0:router(config)# cdp`	Enables CDP globally.
Step 3	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config)# interface pos 0/0/0/1`	Enters interface configuration mode.

	Command or Action	Purpose
Step 4	**cdp** **Example:** `RP/0/RSP0/CPU0:router(config-if)# cdp`	Enables CDP on an interface.
Step 5	**commit**	

Modifying CDP Default Settings

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

Note The commands can be entered in any order.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**cdp advertise v1** **Example:** `RP/0/RSP0/CPU0:router(config)# cdp advertise v1`	Configures CDP to use only version 1 (CDPv1) in communicating with neighboring devices. • By default, when CDP is enabled, the router sends CDPv2 packets. CDP also sends and receives CDPv1 packets if the device with which CDP is interacting does not process CDPv2 packets. • In this example, the router is configured to send and receive only CDPv1 packets.
Step 3	**cdp holdtime** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# cdp holdtime 30`	Specifies the amount of time that the receiving networking device will hold a CDP packet sent from the router before discarding it. • By default, when CDP is enabled, the receiving networking device holds a CDP packet for 180 seconds before discarding it.

	Command or Action	Purpose
		Note The CDP hold time must be set to a higher number of seconds than the time between CDP transmissions, which is set with the **cdp timer** command. • In this example, the value of hold-time for the *seconds* argument is set to 30.
Step 4	**cdp timer** *seconds* **Example:** `RP/0/RSP0/CPU0:router(config)# cdp timer 20`	Specifies the frequency at which CDP update packets are sent. • By default, when CDP is enabled, CDP update packets are sent at a frequency of once every 60 seconds. **Note** A lower timer setting causes CDP updates to be sent more frequently. • In this example, CDP update packets are configured to be sent at a frequency of once every 20 seconds.
Step 5	**commit**	
Step 6	(Optional) **show cdp** **Example:** `RP/0/RSP0/CPU0:router# show cdp`	Displays global CDP information. The output displays the CDP version running on the router, the hold time setting, and the timer setting.

Monitoring CDP

This task shows how to monitor CDP.

Note The commands can be entered in any order.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show cdp entry** {*****	*entry-name*} [**protocol**	**version**] **Example:**	Displays information about a specific neighboring device or all neighboring devices discovered using CDP.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# show cdp entry *`	
Step 2	**show cdp interface** [*type interface-path-id* | **location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show cdp interface pos 0/0/0/1`	Displays information about the interfaces on which CDP is enabled.
Step 3	**show cdp neighbors** [*type interface-path-id* | **location** *node-id*] [**detail**] **Example:** `RP/0/RSP0/CPU0:router# show cdp neighbors`	Displays detailed information about neighboring devices discovered using CDP.
Step 4	**show cdp traffic** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show cdp traffic`	Displays information about the traffic gathered between devices using CDP.

Examples

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

```
RP/0/RSP0/CPU0:router# show cdp neighbors

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

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

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

```
RP/0/RSP0/CPU0:router# show cdp neighbors POS 0/4/0/0 detail

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

  Version :
```

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

  advertisement version: 2
```

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

```
RP/0/RSP0/CPU0:router# show cdp entry router2

  advertisement version: 2

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

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

  advertisement version: 2
```

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

```
RP/0/RSP0/CPU0:router# show cdp interface pos 0/4/0/0

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

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

```
RP/0/RSP0/CPU0:router# show cdp traffic

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

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

```
RP/0/RSP0/CPU0:router# show cdp traffic location 0/4/cpu0

  CDP counters :
          Packets output: 16, Input: 13
```

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

Configuration Examples for Implementing CDP

Enabling CDP: Example

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

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

Modifying Global CDP Settings: Example

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

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

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

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

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

Additional References

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

Related Documents

Related Topic	Document Title
Cisco IOS XR CDP commands	*CDP Commands on Cisco IOS XR Software* module of *System Management Command Reference for Cisco ASR 9000 Series Routers*
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 Cisco IOS XR Software* 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

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 14: 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.

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

What to do next

Note

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

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

Monitoring Periodic MIB Data Collection and Transfer

SUMMARY STEPS

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

DETAILED STEPS

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

show snmp mib bulkstat transfer Sample Output

```
RP/0/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 9

Configuring Flexible Command Line Interface

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

Table 15: 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 106. 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 100 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 10

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 180. 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 16: 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:

```
RP/0/RP0/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 17: Node ID Components, on page 147 summarizes the *node-id* for each type of card.

Table 17: 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 17: Node ID Components, on page 147 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 18: 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.

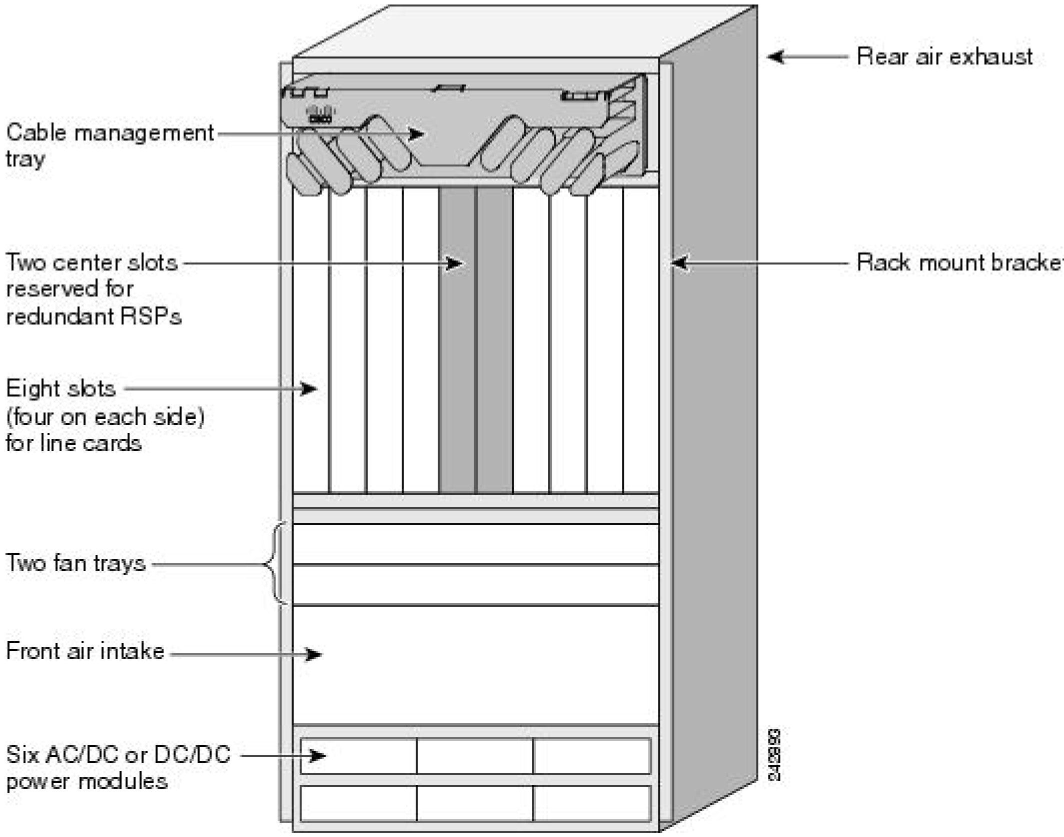
RSP Redundancy and Switchover

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 156). 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 19: RSP Redundancy Commands, on page 157 to display the redundancy status of the cards or force a manual switchover.

Table 19: 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 165

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

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
```

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.

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 162 for information on zero-configuration networking.

For information on CMP shell, see the section CMP Shell, on page 160.

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 20: CMP Shell Commands

Command	Description
adduser	Adds a new CMP user ID/password.

Command	Description
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.
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.

Command	Description
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.
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 21: 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**	
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.

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 22: 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.
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.

Command	Description
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** **Example:** `RP/0/RSP0/CPU0:router# show redundancy`	Displays the RSP redundancy status. • 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.

	Command or Action	Purpose
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 167 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 23: format command Syntax Description, on page 169 describes the **format** command syntax.

Table 23: 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 169 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 169 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 169 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.

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
```

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*
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 11

In Service Software Upgrade for Cisco IOS XR 64 Bit

This module contains the following topics:

Overview of ISSU on ASR 9000 with IOS XR 64 Bit

In-Service Software Upgrade (ISSU) provides the ability to upgrade the IOS XR 64 Bit version on ASR 9000 with minimal disruption on the control plane and forwarding plane. ISSU supports upgrading an image from a lower to a higher version. ISSU supports zero topology loss (ZTL) and causes only a minimal packet loss of less than six seconds.

You can perform ISSU installation in a single step or as multiple phases. You need to perform the pre-installation tasks before executing ISSU. During the pre-installation tasks and ISSU execution, V1 refers to the image currently running on the router and V2 will be the upgraded image.

ISSU execution contains the following phases:

- Prepare phase: The installable files are pre-checked and loaded on the router before activation. This phase is optional.
- Activate phase: The new image (V2) is downloaded to all nodes in the router replacing the old image (V1). This phase can be run in step-by-step phases like *Load*, *Run*, and *Cleanup* or by using a one-shot *Activate* phase.

Note The *Prepare* phase is optional and can be skipped because the *Load* phase prepares the package if *Prepare* phase was not performed before the *Load* phase.

- Commit phase: The ISSU installation is complete with V2 on all nodes.

ISSU supports upgrading the System Admin VM and XR VM individually. Using ISSU, the System Admin VM and XR VM can also be upgraded sequentially. The upgrade sequence is System Admin ISSU followed by XR ISSU. Committing the upgrade from XR VM commits both the System Admin and XR software. But, committing the upgrade from System Admin VM commits only the System Admin software.

Note When RSP1 is the active RP and System Admin VM ISSU is triggered on Cisco ASR 9000 with IOS XR 64 bit, there is an additional VM switch over compared to performing System Admin VM ISSU from RSP0. This is an expected behaviour.

Restrictions and Usage Guidelines

ISSU on ASR 9000 with IOS XR 64 Bit is supported only on few third generation ASR 9000 Series Ethernet line cards, RSPs and RPs.

The third generation of line cards include:

- A9K-8X100G-LB-SE
- A9K-8X100G-LB-TR
- A9K-8X100GE-SE
- A9K-8X100GE-TR
- A9K-4X100GE-SE
- A9K-4X100GE-TR
- A9K-MOD400-SE
- A9K-MOD400-TR
- A9K-MOD200-SE
- A9K-MOD200-TR
- A9K-4X100GE
- A99-12X100GE
- A99-12X100GE-CM
- A9K-24X10GE-1G-SE
- A9K-24X10GE-1G-TR
- A9K-48X10GE-1G-SE
- A9K-48X10GE-1G-TR
- A99-48X10GE-1G-SE
- A99-48X10GE-1G-TR

The third generation of RSP and RP cards include:

- A9K-RSP880-SE/TR
- A9K-RSP880-LT-SE/TR
- A99-RSP-SE/TR
- A99-RP2-SE/TR

ISSU is not supported on the ASR 9000 fourth generation QSFP28 based dense 100GE line cards. The line cards include:

- A9K-16X100GE-TR
- A99-32X100GE-TR
- A99-16X100GE-X-SE

ISSU is not supported on the single RP system of ASR 9901.

Pre-Installation Tasks

Pre-Installation Tasks

Before performing ISSU on ASR 9000 with IOS XR 64 Bit, complete the following tasks.

1. Configure NTP in XR VM. Once you configure NTP on XR VM, System Admin VM automatically syncronizes with NTP running on RSP. If NTP server is not available, configure clock on both XR VM and System Admin VM in configuration mode and make sure that your clock is set to the correct location and timezone.

```
RP/0/RSP0/CPU0:Router# show ntp associations
      address         ref clock     st  when  poll reach  delay  offset    disp
*~172.27.131.19    171.68.38.65      2    13    64   77    2.05  -1.174  191.09
 ~172.27.130.34    171.68.38.65      2    35    64    0    0.00   0.000   15937
+~172.27.130.33    171.68.38.65      2     9    64   77    2.41  10.370  189.47
 * sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

```
sysadmin-vm:0_RSP0:Router# show ntp associations
Wed Oct  31 15:18:59.728 UTC-07:00
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
external:
*rsp0_xr         172.27.131.19    3 u   12  128  347    0.171    0.989   0.456
 rsp1_xr         .STEP.          16 u    -  256    0    0.000    0.000   0.000
internal:
*192.0.16.4      172.27.131.19    3 u   12  128  347    0.171    0.989   0.456
 192.0.20.4      .STEP.          16 u    -  256    0    0.000    0.000   0.000
```

2. Ensure that the dual RP and RSP systems are synchronized and they are in active and standby roles respectively using the **show redundancy summary** command. The line card status should be *Final Band* or *Running*.

```
RP/0/RSP0/CPU0:Router# show redundancy summary

    Active Node    Standby Node
    -----------    ------------
    0/RSP0/CPU0     0/RSP1/CPU0  (Node Ready, NSR:Ready)
```

```
RP/0/RSP0/CPU0:Router# show platform vm
Node name       Node type       Partner name    SW status       IP address
--------------- --------------- --------------- --------------- ---------------
0/3/CPU0        LC (ACTIVE)     NONE            FINAL Band      192.0.12.3
0/7/CPU0        LC (ACTIVE)     NONE            FINAL Band      192.0.36.3
0/RSP1/CPU0     RP (STANDBY)    0/RSP0/CPU0     FINAL Band      192.0.20.4
0/4/CPU0        LC (ACTIVE)     NONE            FINAL Band      192.0.24.3
0/RSP0/CPU0     RP (ACTIVE)     0/RSP1/CPU0     FINAL Band      192.0.16.4
```

3. Ensure that firmware on linecards, RSP, and RP is upgraded to the latest version. You can upgrade the router cards in a single step by using the **upgrade hw-module location all fpd all** command. Use the **show hw-module location** *location* **fpd** command to verify the firmware versions.

```
RP/0/RSP0/CPU0:Router# upgrade hw-module location all fpd all
```

This example shows verifying the firmware versions for a node.

```
RP/0/RSP0/CPU0:Router# show hw-module location 0/rsp1 fpd
                                                               FPD Versions
                                                               =================
Location   Card type        HWver FPD device       ATR Status   Running Programd
------------------------------------------------------------------------------
0/RSP1    A9K-RSP880-SE     1.0   Alpha-FPGA           CURRENT    0.16    0.16
0/RSP1    A9K-RSP880-SE     1.0   CBC                  CURRENT   34.39   34.39
0/RSP1    A9K-RSP880-SE     1.0   Cha-FPGA             CURRENT    0.08    0.08
0/RSP1    A9K-RSP880-SE     1.0   IPU-FPGA             CURRENT    0.66    0.66
0/RSP1    A9K-RSP880-SE     1.0   IPU-FSBL             CURRENT    1.108   1.108
0/RSP1    A9K-RSP880-SE     1.0   IPU-Linux            CURRENT    1.108   1.108
0/RSP1    A9K-RSP880-SE     1.0   Omega-FPGA           CURRENT    0.16    0.16
0/RSP1    A9K-RSP880-SE     1.0   Optimus-FPGA         CURRENT    0.12    0.12
0/RSP1    A9K-RSP880-SE     1.0   Primary-BIOS         CURRENT   10.60   10.60
0/RSP1    A9K-RSP880-SE     1.0   SSDa-SMART           N/A        7.05    7.05
0/RSP1    A9K-RSP880-SE     1.0   SSDb-SMART           N/A        7.05    7.05
```

4. Check the disk storage space on both System Admin VM and XR VM and ensure that sufficient disk space is available. Remove files like show-tech, cores, kernel dumps, manually created text, log, debug information and so on.

This example shows verifying the disk storage space for System Admin VM and XR VM on RSP0. You also need to verify the disk space on the standby RSP (RSP1). If required you can verify the disk storage on line cards using the **show media location** command.

```
RP/0/RSP0/CPU0:Router# show media

Media Information for local node.

----------------------------------------------
Partition                    Size     Used  Percent    Avail
rootfs:                      3.8G     1.3G      34%     2.4G
/run                          14G     340K       1%      14G
harddisk:                    5.5G     1.6G      31%     3.6G
/run/netns                    14G     340K       1%      14G
log:                         469M      33M       8%     401M
config:                      469M     1.7M       1%     432M
disk0:                       968M     5.3M       1%     897M
harddiska:                   3.6G      11M       1%     3.5G
/misc/app_host               2.4G      61M       3%     2.2G
---------------------------------------------------
rootfs: = root file system (read-only)
log: = system log files (read-only)
config: = configuration storage (read-only)

RP/0/RSP0/CPU0:Router# admin show media
---------------------------------------------------------------------
```

```
Partition                    Size     Used  Percent    Avail
--------------------------------------------------------------------
rootfs:                      2.4G     838M      38%     1.4G
harddisk:                    7.6G     1.2G      17%     6.0G
log:                         469M      22M       5%     412M
config:                      469M     3.3M       1%     431M
disk0:                       968M     1.6M       1%     900M
a9ksys:                      736M     4.3M       1%     695M
harddiskb:                   3.0G     1.2G      41%     1.8G
install:                     4.8G     932M      21%     3.6G
install:/tmp                 7.6G     1.2G      17%     6.0G
install:/cache               7.6G     1.2G      17%     6.0G
rootfs:/install/tmp          7.6G     1.2G      17%     6.0G
--------------------------------------------------------------------
rootfs:  = root file system (read-only)
log:     = system log files (read-only)
config:  = configuration storage (read-only)
install: = install repository (read-only)
a9ksys:  = ASR9K system storage (read-only)
```

5. Populate the repository with RPMs and SMUs. You can pick and install individual RPMs, SMUs, one by one, or make a tarball and install one tarball or break it down with multiple tarballs.

Note You cannot include a tarball within another tarball. However, multiple tarballs can be specified at once.

```
RP/0/RSP0/CPU0:Router# install add source tftp://172.27.131.19/xrimages/e652/
asr9k-mini-x64-6.5.2.13I.iso  ASR9K-RPMS-65213I.tar

RP/0/RSP0/CPU0:Router# show install request
The install add operation 4 is 80% complete
RP/0/RSP0/CPU0:Router#

RP/0/RSP0/CPU0:Router#
RP/0/RSP0/CPU0:Oct 31 19:44:34.076 : sdr_instmgr[1156]:
%INSTALL-INSTMGR-2-OPERATION_SUCCESS : Install operation 4 finished successfully
Oct 31 19:44:35 Install operation 4 finished successfully
```

6. Check the repository to validate that packages, images, or SMUs are populated properly in the router's repository by using the**show install repository** command. There should be a one to one relationship between V1 and V2 images and SMUs. For example, if you install a SMU on V1, you also need the corresponding V2 version in the repository to execute ISSU.

```
RP/0/RSP0/CPU0:Router# show install repository | i mini
asr9k-mini-x64-6.2.3                   <--V1 iso image currently running
asr9k-mini-x64-6.5.2.13I               <--V2 iso image to upgrade to
```

7. Extract the ISO image in System Admin VM or XR VM depending on the version of the image.

   - For IOS XR versions prior to 6.5.1: You should extract the ISO image in XR VM as well as in System Admin VM because the system can only use packages in RPM format.
   - For IOS XR version 6.5.1 and onwards: System automatically extracts the system admin package from the ISO file once you mentioned the file name while executing ISSU. You only need to extract the XR package separately.

```
RP/0/RSP0/CPU0:Router# install extract asr9k-mini-x64-6.5.2.13I
Oct 31 20:50:30 Install operation 9 started by root:
  install extract asr9k-mini-x64-6.5.2.13I
Oct 31 20:50:30 Package list:
```

```
Oct 31 20:50:30     asr9k-mini-x64-6.5.2.13I
Oct 31 20:50:31 Install operation will continue in the background

RP/0/RSP0/CPU0:Router# show install repository | i xr-

    asr9k-xr-6.5.2.13I                          <--V2 extracted image to upgrade to
    asr9k-xr-6.2.3

sysadmin-vm:0_RSP0:Router# install extract asr9k-mini-x64-6.5.2.13I
result Wed Oct 31 20:36:34 2018 Install operation 1 (install extract) started by user
'root' will continue asynchronously.
sysadmin-vm:0_RSP0:MYISSU#

sysadmin-vm:0_RSP0:Router# show install repository | i sys
asr9k-sysadmin-6.2.3
 asr9k-sysadmin-6.5.2.13I                       <--V2 extracted image to upgrade to
```

ISSU on ASR 9000 with IOS XR 64 Bit: Single Phase Installation

This section shows how to perform ISSU on ASR 9000 with IOS XR 64 Bit in a single step. You can either upgrade the system or install a patch in a single step. The system upgrade is done using an ISO image file, while the patch installation is done using packages and SMUs.

You should perform the following steps before performing this task:

- Copy the package to be installed either on the router's hard disk or on a network server to which the router has access.
- Ensure that dual route processor (RP) system with standby is in "is ready" state.

Preform the following steps to upgrade the system or install a patch in a single step.

Note Depending on whether you are installing a System Admin package or a XR package, execute these commands in the System Admin EXEC mode or XR EXEC mode respectively

1. (Optional) Prepare the installable files by using the **install prepare issu** *package_name* command. During the prepare phase, pre-activation checks are made, and the components of the installable files are loaded on to the router setup.

   For System Admin VM:

   ```
   sysadmin-vm:0_RSP0# install prepare issu asr9k-xr-<release-version>
   ```

   For XR VM:

   ```
   RP/0/RSP0/CPU0:router# install prepare issu asr9k-xr-<release-version>
   ```

2. Activate the ISSU installation in XR VM or System Admin VM in a single step by using the **install activate issu** command.

   For System Admin VM:

   ```
   sysadmin-vm:0_RSP0# install activate issu asr9k-xr-<release-version>
   ```

   For XR VM:

   RP/0/RSP0/CPU0:router# `install activate issu asr9k-xr-<release-version>`

Note ISSU operation takes about 30 minutes to complete. If the ISSU operation is not concluded in 40 minutes, the ISSU may timeout or abort.

3. Commit the newly active software by using the **install commit** command.

   For System Admin VM:

   ```
   sysadmin-vm:0_RSP1# install commit
   ```

   For XR VM:

   ```
   RP/0/RSP0/CPU0:router# install commit
   ```

Examples: Install Packages Using ISSU Single Step Installation on ASR 9000 with IOS XR 64 Bit

This example shows performing System Admin VM upgrade using ISSU and how to verify the installation using show commands.

```
!# Verify packages in the repository

sysadmin-vm:0_RSP0:Router# show install repository all | i "host|sysadmin"
asr9k-sysadmin-6.2.3
 asr9k-sysadmin-6.5.2.13I
 host-6.2.3
 host-6.5.2.13I
sysadmin-vm:0_RSP0:Router#

!# Performing ISSU installation

sysadmin-vm:0_RSP0:Router# install activate issu asr9k-sysadmin-6.5.2.13I host-6.5.2.13I
This install operation will result in admin VMs reload
Do you want to proceed [yes/no]: yes
Proceeding with operation
result Wed Oct 31 21:12:21 2018 Install operation 2 (install prepare and activate issu)
started by user 'root' will continue asynchronously.
sysadmin-vm:0_RSP0:Router#

!# Monitoring the progress of the installation.
!# The installation may take up to 30 minutes.

sysadmin-vm:0_RSP0:Router# show install request

User root, Op Id 2

 install prepare issu

 host-6.5.2.13I

 This operation is 40% complete

 Waiting for agents to complete host prepare ..

sysadmin-vm:0_RSP0:#


!# Verifying the installation status again after few minutes

sysadmin-vm:0_RSP0:Router# show install request
```

```
User root, Op Id 2
 install activate issu
 ISSU stage Phase1
 asr9k-sysadmin-6.5.2.13I
   Node 0/RSP0 [RP] : 90% of current state is completed
   Node 0/RSP1 [RP] : 90% of current state is completed
   Node 0/1 [LC] : 90% of current state is completed
   Node 0/3 [LC] : 90% of current state is completed
   Node 0/4 [LC] : 90% of current state is completed
   Node 0/7 [LC] : 90% of current state is completed

sysadmin-vm:0_RSP0:MYISSU#

!#  Message after successful completion. Admin VM will reload after this message. . There
should be no packet drop.

0/RSP0/ADMIN0:Oct 31 21:27:53.260 : inst_mgr[5019]: %INFRA-INSTMGR-2-OPERATION_SUCCESS :
Install operation 2 completed successfully

!# Verifying the active package

sysadmin-vm:0_RSP1# show install active summary
     Active Packages: 1
        asr9k-sysadmin-6.5.2.13I version=6.5.2.13I [Boot image]

!# Verifies the image previously committed

sysadmin-vm:0_RSP1# show install commit summary
     Committed Packages: 1
        asr9k-sysadmin-6.2.3 version=6.2.3 [Boot image]

!# Commits the latest image

sysadmin-vm:0_RSP1# install commit
result Wed Oct 31 21:32:58 2018 Install operation 3 (install commit) started by user 'root'
 will continue asynchronously.
sysadmin-vm:0_RSP1# 0/RSP1/ADMIN0:Oct 31 21:33:02.061 : inst_mgr[6913]:
%INFRA-INSTMGR-2-OPERATION_SUCCESS : Install operation 3 completed successfully
Wed Oct 31 21:33:02 2018 Install operation 3 completed successfully.
sysadmin-vm:0_RSP1#
```

This example shows performing XR VM upgrade using ISSU and verifying the installation using show commands.

```
!# Verify the active packages

RP/0/RSP0/CPU0:Router# show install active summary
    Active Packages: 8
        asr9k-xr-6.2.3 version=6.2.3 [Boot image]
        asr9k-isis-x64-1.3.0.0-r623
        asr9k-mpls-x64-2.0.0.0-r623
        asr9k-mpls-te-rsvp-x64-1.3.0.0-r623
        asr9k-ospf-x64-1.0.0.0-r623
        asr9k-mgbl-x64-2.0.0.0-r623
        asr9k-mcast-x64-2.2.0.0-r623
        asr9k-k9sec-x64-3.2.0.0-r623

!# Performing ISSU Installation

RP/0/RSP0/CPU0:Router# install activate issu asr9k-xr-6.5.2.13I *r65213I
Oct 31 21:48:14 Install operation 10 started by root:
  install activate issu asr9k-xr-6.5.2.13I *r65213I
Oct 31 21:48:14 Package list:
```

```
Oct 31 21:48:14     asr9k-isis-x64-1.1.0.0-r65213I.x86_64
Oct 31 21:48:14     asr9k-ospf-x64-1.0.0.0-r65213I.x86_64
Oct 31 21:48:14     asr9k-mcast-x64-2.0.0.0-r65213I.x86_64
Oct 31 21:48:14     asr9k-mpls-te-rsvp-x64-2.1.0.0-r65213I.x86_64
Oct 31 21:48:14     asr9k-mgbl-x64-2.0.0.0-r65213I.x86_64
Oct 31 21:48:14     asr9k-k9sec-x64-2.1.0.0-r65213I.x86_64
Oct 31 21:48:14     asr9k-mpls-x64-2.0.0.0-r65213I.x86_64
Oct 31 21:48:14     asr9k-xr-6.5.2.13I
This install operation will start the issu, continue?
 [yes/no]:[yes] yes
Oct 31 21:49:13 Install operation will continue in the background
RP/0/RSP0/CPU0:Router#

!# Monitoring the progress of the installation.
!# The installation may take up to 30 minutes.

RP/0/RSP1/CPU0:Router# show issu
INSTALL Operation ID    : Operation 11 Started at Wed Oct 31 22:23:30 2018
ISSU Progress           : 100.0%
Total ISSU Time         : 00:25:07
ISSU Type               : SMU

Phase           Start-Time      End-Time                  State
--------------------------------------------------------------------------------
Prepare         22:23:30        22:34:21                  Completed
Load            22:34:28        22:47:01                  Completed
Run             22:47:01        22:47:44                  Completed
Cleanup         22:47:44        22:48:44                  Completed
--------------------------------------------------------------------------------
Current Status          : ISSU Orchestration Successfully Completed

Setup Information       : Single Chassis
ISSU Ready/Not Ready    : 0 / 0

Node ISSU readiness per rack per slot
Key: Ready - 'Y', Not ready - 'N', Primary node - '*', Complete - '-'

Rack 0   RP0   RP1   LC1   LC3   LC4   LC7
          -     -     -     -     -     -

!# Verifying the VM status after the installation

RP/0/RSP1/CPU0:Router# admin show sdr

SDR: default-sdr
Location     IP Address      Status           Boot Count  Time Started
-----------------------------------------------------------------------------
0/1/VM2      192.0.4.4       RUNNING          1           10/31/2018 22:34:55
0/3/VM2      192.0.12.4      RUNNING          1           10/31/2018 22:34:54
0/RSP0/VM2   192.0.16.6      RUNNING          1           10/31/2018 22:49:58
0/RSP1/VM2   192.0.20.6      RUNNING          1           10/31/2018 22:35:39
0/4/VM2      192.0.24.4      RUNNING          1           10/31/2018 22:34:55
0/7/VM2      192.0.36.4      RUNNING          1           10/31/2018 22:34:55

!# Verify the active packages

RP/0/RSP1/CPU0:Router# show install active sum
    Active Packages: 8
        asr9k-xr-6.5.2.13I version=6.5.2.13I [Boot image]
        asr9k-isis-x64-1.1.0.0-r65213I
        asr9k-ospf-x64-1.0.0.0-r65213I
        asr9k-mcast-x64-2.0.0.0-r65213I
        asr9k-mpls-te-rsvp-x64-2.1.0.0-r65213I
        asr9k-mgbl-x64-2.0.0.0-r65213I
```

```
        asr9k-k9sec-x64-2.1.0.0-r65213I
        asr9k-mpls-x64-2.0.0.0-r65213I


!# You can either perform install commit and stay with the latest image or reload the router
 to continue using the old image.

!# Commits the latest image after the necessary checks
RP/0/RSP0/CPU0:Router# install commit
```

ISSU on ASR 9000 with IOS XR 64 Bit: Multi Step Installation

This section shows how to perform ISSU on ASR 9000 with IOS XR 64 Bit in multiple steps.

You should perform the following steps before performing the steps in this task:

- Copy the package to be installed either on the router's hard disk or on a network server to which the router has access.
- Ensure that dual route processor (RP) system with standby is in "is ready" state.

Perform the following steps to upgrade the system or install a patch in multiple phases.

Note Depending on whether you are installing a System Admin package or a XR package, execute these commands in the System Admin EXEC mode or XR EXEC mode respectively.

Note You should update the System Admin VM first and then update the XR VM. IOS XR 64 bit ISSU will fail if the System Admin VM is not updated first.

1. (Optional) Prepare the installable files by using the **install prepare issu** *package_name* command. During the prepare phase, pre-activation checks are performed and the components of the installable files are loaded on to the router setup.

   For System Admin VM :

   ```
   sysadmin-vm:0_RSP0# install prepare issu asr9k-xr-<release-version>
   ```

   For XR VM:

   ```
   RP/0/RSP0/CPU0:router# install prepare issu asr9k-xr-<release-version>
   ```

2. Start the load phase by issuing the **install activate issu load** command.

   For System Admin VM :

   ```
   sysadmin-vm:0_RSP0# install activate issu load asr9k-xr-<release-version>
   ```

   For XR VM:

   ```
   RP/0/RSP0/CPU0:router# install activate issu load asr9k-xr-<release-version>
   ```

   This step downloads the new image (V2) to all nodes in the router. The new image is checked for compatibility to ensure that the router can be upgraded. At the start of the *Load* phase, the router configuration mode is locked, and you cannot perform any configuration on the router until ISSU completes the phase. At the end of this stage, all standby nodes run V2 and all active nodes (including all line cards) still run the original software images (V1).

3. Starts the run phase by issuing the **install activate issu run** command.

   For System Admin VM :

   ```
   sysadmin-vm:0_RSP0# install activate issu run
   ```

   For XR VM:

   ```
   RP/0/RSP0/CPU0:router# install activate issu run
   ```

   This phase starts version switch from V1 to V2. All the packages that have been prepared are activated to make the package configurations active on the router.

4. Starts the cleanup phase by issuing the **install activate issu cleanup** command.

   For System Admin VM :

   ```
   sysadmin-vm:0_RSP0# install activate issu cleanup
   ```

   For XR VM:

   ```
   RP/0/RSP0/CPU0:router# install activate issu cleanup
   ```

   This phase Initiates shutdown of VMs with previous versions after running the activation. The cleanup phase concludes the ISSU process and the new software runs on all nodes in the system.

5. Commit the newly active software by using the **install commit** command.

   For System Admin VM:

   ```
   sysadmin-vm:0_RSP0# install commit
   ```

   For XR VM:

   ```
   RP/0/RSP0/CPU0:router# install commit
   ```

Installing Packages Using ISSU : Related Commands

Related Commands	Purpose
show install active	Displays the active packages on the system.
show install request	Displays the progress of the ISSU installation.
show issu	Displays the state or status of the ISSU operation. Effective with Cisco IOS XR version 6.5.1, this command is also supported for System Admin VM ISSU.
install prepare clean	Clears the existing prepared image. If there is a failure in the prepare phase, you can run this command to clear the prepared image.
install activate ISSU abort	Initiates ISSU abort in XR VM. ISSU aborts if the command is run before ISSU Run phase starts. All the changes due to the install activity are reset.

CHAPTER 12

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 24: 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

Restriction: 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.

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.

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.

Note 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.

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**.

Note CFP2-DCO Optical modules do not support automatic-FPD upgrade.

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:

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.

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)
```

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.

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.)

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td></td><td>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.</td></tr>
<tr><td>Step 4</td><td>upgrade hw-module fpd {all | fpga-type} [force] location [all | node-id]

Example:
<pre>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</pre></td><td>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:
<pre>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.</pre>
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:
<pre>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)]</pre>
If you confirm that you want to abort the FPD upgrade procedure, this message is displayed:
<pre>FPD upgrade process has been aborted, please
check the status of the hardware and reissue
the upgrade command if required.</pre></td></tr>
</table>

	Command or Action	Purpose
		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    fpqa1    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 25: 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.
Type	Hardware type. Can be one of the following types: • spa—Shared port adapter • lc—Line card

Field	Description
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.

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 26: 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 13

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 237. 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 27: 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 228

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.

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:** `RP/0/RSP0/CPU0:router(config-ntp)# interface POS 0/1/0/0`	Enters NTP interface configuration mode.
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.

	Command or Action	Purpose
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:
	`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.

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.
		• 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.

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* • **interface** *type interface-path-id* **disable** **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# no interface pos 0/0/0/1` or	Disables NTP services on the specified interface.

	Command or Action	Purpose
	`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 216.

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 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.
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-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 406 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

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 1</td><td>configure
Example:
<code>RP/0/RSP0/CPU0:router# configure</code></td><td>Enters global configuration mode.</td></tr>
<tr><td>Step 2</td><td>ntp
Example:
<code>RP/0/RSP0/CPU0:router(config)# ntp</code></td><td>Enters NTP configuration mode.</td></tr>
<tr><td>Step 3</td><td>master primary-reference-clock
Example:
<code>RP/0/RSP0/CPU0:router(config-ntp)# master primary-reference-clock</code></td><td>Specifies PTP to be the NTP time source.</td></tr>
<tr><td>Step 4</td><td>Use one of the following commands:
• end
• commit
Example:
<code>RP/0/RSP0/CPU0:router(config-ntp)# end</code>
or
<code>RP/0/RSP0/CPU0:router(config-ntp)# commit</code></td><td>Saves configuration changes.
• When you issue the end command, the system prompts you to commit changes:
<code>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</code>
• 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.</td></tr>
</table>

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** • **commit** **Example:** `RP/0/RSP0/CPU0:router(config-ntp)# end` or	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]:`

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-ntp)# commit`	• 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

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

Configure FQDN for NTP server

Configuration Example for FQDN on NTP Server on Default VRF

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

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

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

Running Configuration

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

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

Verification

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

```
Router#show ntp associations

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

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.
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.

	Command or Action	Purpose
		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 216.

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*

Related Topic	Document Title
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 14

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 242
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 28: Related Documents

Related Topic	Document Title
Netconf-Yang	For related commands, see *System Security Command Reference for Cisco ASR 9000 Series Routers*

Table 29: Standards

Component	RFCs
YANG	6020
NETCONF	6241
NETCONF over SSH	6242

CHAPTER 15

Configuring Disk Mirroring

This module describes the process to configure disk mirroring in Cisco IOS XR software.

For complete descriptions of the commands listed in this module, see Related Documents, on page 256. 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 30: Feature History for Disk Mirroring for Cisco IOS XR Software

Release	**Modification**
Release 3.7.2	Disk mirroring was introduced.
Release 3.9.0	No modification.

This module contains the following topics:

Disk Mirroring Prerequisites

Before enabling disk mirroring, the following conditions must be met:

- 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.
- The secondary storage device specified for the mirroring must be installed in the same node as the primary boot device. The supported storage devices are disk0: and disk1:.
- The secondary storage device must be the same size or larger than the designated primary storage device.
- The secondary storage device must be partitioned.

Note The primary partition on the secondary storage device must be large enough to contain all data on the primary boot device. This can be an issue if the primary boot device has not yet been partitioned. For example, in the situation where both the primary boot device and the secondary storage device are 1 GB in size, the primary boot device contains 950 MB of data, and the secondary storage device is already partitioned to 800 MB in the primary partition and 200 MB in the secondary partition. In such a case, the 950 MB of data from the primary boot device does not fit on the secondary storage device because of the partition. Such a configuration is rejected and an error is displayed. You need to replace the secondary storage device with a higher capacity device. For information about disk partition sizes, see *Related Topics*.

Note Although compactflash: can be used as the secondary device on a Performance Route Processor (PRP–2), there is an issue with the ROM Monitor not being able to boot the minimum boot image (MBI) from the secondary device if the device is not disk0: or disk1:. In such a situation, you would need to go into ROMMON mode and boot the PRP-2 manually using the MBI on the compactflash:.

Related Topics

Information About Disk Mirroring, on page 248

Information About Disk Mirroring

The route switch processor (RSP) card has a primary storage device that is used to store installation packages and configuration files. This primary storage device is referred to as the *primary boot device* and is essential for booting the RSP and its normal operation.

Disk mirroring replicates the critical data on the primary boot device onto another storage device on the same RSP, henceforth referred to as the secondary device. If the primary boot device fails, applications continue to be serviced transparently by the secondary device, thereby avoiding a switchover to the standby RSP. The failed primary storage device can be replaced or repaired without disruption of service.

Disk mirroring should only mirror critical data on the primary boot device onto a secondary storage device and not any noncritical data such as logging data. To separate critical data from noncritical data, the disk devices need to be partitioned. Disk0: is partitioned to disk0: and disk0a:; disk1: is partitioned to disk1: and disk1a:. Disk0: and disk1: are used for critical data, whereas disk0a: and disk1a: are used for logging data and other noncritical data. Before you can configure disk mirroring on the RSP, you must have partitioned the secondary storage device. The sizes of disk partitions are related to the total disk size, and are provided in Table 31: Size of Disk Partitions in Relation to Size of Disk, on page 248.

Table 31: Size of Disk Partitions in Relation to Size of Disk

Size of Disk	Primary Partition Percentage	Secondary Partition Percentage
less than 900 MB	Partitioning not supported	Partitioning not supported
900 MB to 1.5 GB	80%	20%
1.5 GB to 3 GB	60%	40%
more than 3 GB	50%	50%

How to Enable Disk Mirroring

The tasks in this section describe how to enable and manage disk mirroring.

Enabling Disk Mirroring

Complete the following instructions to enable disk mirroring. After disk mirroring is configured, if there is a fault on the primary boot drive or it cannot be accessed for any reason, control is automatically transferred to the secondary storage device.

SUMMARY STEPS

1. **format** *secondary-device* **partition** [**location** *node-id*]
2. Remove any noncritical data from the primary boot device.
3. **configure**
4. **mirror location** *node-id* *Primary-device Secondary-device*
5. Use the **commit** or **end** command.
6. **show mirror** [**location** *node-id*]
7. **mirror verify location** *node-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**format** *secondary-device* **partition** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# format disk1: partition`	Partitions the secondary storage device into two partitions. • If the device is already partitioned, you do not need to perform this step.
Step 2	Remove any noncritical data from the primary boot device.	The primary boot device should contain installation packages and configuration files only. Log files can be copied to the "a" partition of the secondary device, for example disk1a: .
Step 3	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 4	**mirror location** *node-id* *Primary-device Secondary-device* **Example:** `RP/0/RSP0/CPU0:router(config)# mirror location 0/rsp0/cpu0 disk0:disk1:`	Enables disk mirroring of the *primary-device* to the *secondary-device* . If the primary boot device is not partitioned, the following occurs: • The contents of the primary device are replicated to the secondary device

	Command or Action	Purpose
		• Control of the mirroring server switches to the secondary storage device. • The primary device is partitioned. • Data is replicated back to the primary boot device.
Step 5	Use the **commit** or **end** command.	**commit** —Saves the configuration changes and remains within the configuration session. **end** —Prompts user to take one of these actions: • **Yes** — Saves configuration changes and exits the configuration session. • **No** —Exits the configuration session without committing the configuration changes. • **Cancel** —Remains in the configuration session, without committing the configuration changes.
Step 6	**show mirror** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mirror location 0/rsp0/cpu0`	Displays disk mirroring information for an RSP node. It also provides the status of the synchronization between the primary and secondary devices.
Step 7	**mirror verify location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# mirror verify location 0/rsp0/cpu0`	Verifies disk synchronization for disk mirroring on an RSP node.

Replacing the Secondary Mirroring Device

Follow this procedure if you need to replace the secondary boot device used in the disk mirroring process.

SUMMARY STEPS

1. **show mirror** [**location** *node-id*]
2. **mirror pause** [**location** *node-id*]
3. **show mirror** [**location** *node-id*]
4. **unmount** *secondary-device* [**location** *node-id*]
5. Remove the device and insert a new device.
6. **format** *secondary-device* **partition** [**location** *node-id*]
7. **show media** [**location** *node-id*]
8. **mirror resume** [**location** *node-id*]
9. **show mirror** [**location** *node-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show mirror** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mirror`	Verifies that mirroring is active. In the output, the *Current Mirroring State* should be redundant.
Step 2	**mirror pause** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# mirror pause`	Temporarily pauses disk mirroring.
Step 3	**show mirror** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mirror`	Verifies that mirroring has paused. In the output, the *Current Mirroring State* should be paused.
Step 4	**unmount** *secondary-device* [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# unmount disk1:`	Unmounts the secondary device.
Step 5	Remove the device and insert a new device.	
Step 6	**format** *secondary-device* **partition** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# format disk1: partition`	Formats the device.
Step 7	**show media** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show media`	Verifies that the device is formatted. The output should display the device that you formatted.
Step 8	**mirror resume** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# mirror resume`	Resumes mirroring.
Step 9	**show mirror** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mirror`	Verifies that mirroring has restarted. In the output, the *Current Mirroring State* should be Syncing. It can take 15 to 30 minutes for the mirroring process to complete. The exact time depends on the number of packages or files on the boot device. When the mirroring is complete, the *Current Mirroring State* should be Redundant.

Replacing the Primary Mirroring Device

In the event that your primary boot disk is defective and you need to replace it while disk mirroring is enabled, perform this task.

SUMMARY STEPS

1. **show mirror** [**location** *node-id*]
2. **configure**
3. **mirror location** *node-id Primary-device Secondary-device*
4. Use the **commit** or **end** command.
5. **show mirror** [**location** *node-id*]
6. **mirror pause** [**location** *node-id*]
7. **show mirror**
8. **unmount** *secondary-device* [**location** *node-id*]
9. Remove the device and insert a new device.
10. **show media** [**location** *node-id*]
11. (Optional) **format** *secondary-device* **partition** [**location** *node-id*]
12. **mirror resume** [**location** *node-id*]
13. **show mirror** [**location** *node-id*]
14. **configure**
15. **mirror location** *node-id Primary-device Secondary-device*
16. **show mirror** [**location** *node-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show mirror** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mirror`	Verifies that mirroring is in the redundant state. In the output, the *Current Mirroring State* should be redundant. If mirroring is not in the redundant state, you cannot proceed with the procedure. You must wait until mirroring is in the redundant state.
Step 2	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

Step 3	**mirror location** *node-id Primary-device Secondary-device*
	Example:
	`RP/0/RSP0/CPU0:router(config)# mirror location 0/`
	`RSP0`
	`/CPU0 disk1:disk0:`
		Swaps the device roles such that the primary mirroring device now becomes the secondary device and the secondary mirroring device becomes the primary device.

	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 session, without committing the configuration changes.
Step 5	**show mirror** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mirror`	Verifies that the primary device is now the secondary device and vice versa. In the output, if disk0: was the primary disk that you want to replace, it should now be listed as the secondary device.
Step 6	**mirror pause** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# mirror pause`	Temporarily pauses disk mirroring.
Step 7	**show mirror** **Example:** `RP/0/RSP0/CPU0:router# show mirror`	Verifies that mirroring has paused. In the output, the *Current Mirroring State* should be paused.
Step 8	**unmount** *secondary-device* [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# unmount disk1:`	Unmounts the secondary device which is the device that you want to replace. Initially, this was the primary device.
Step 9	Remove the device and insert a new device.	
Step 10	**show media** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show media`	Verifies that the new disk is partitioned. You should see that the new device is mounted. If the new device is not partitioned, format the device as indicated in the next step.
Step 11	(Optional) **format** *secondary-device* **partition** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# format disk1: partition`	Formats the device. You only need to perform this step if the new device is not partitioned.
Step 12	**mirror resume** [**location** *node-id*] **Example:**	Resumes mirroring.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# mirror resume`	
Step 13	**show mirror** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mirror`	Verifies that mirroring has restarted. In the output, the *Current Mirroring State* should be Syncing. It can take 15 to 30 minutes for the mirroring process to complete. The exact time depends on the number of packages or files on the boot device. When the mirroring is complete, the *Current Mirroring State* should be Redundant.
Step 14	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.

Step 15	**mirror location** *node-id Primary-device Secondary-device*
	Example:
	`RP/0/RSP0/CPU0:router(config)# mirror location 0/`
	`RSP0`
	`/CPU0 disk0:disk1:`
		Swaps the device roles back so that the newly inserted device becomes the primary device.

Step 16	**show mirror** [**location** *node-id*] **Example:** `RP/0/RSP0/CPU0:router# show mirror`	Verifies that the new device is now the primary device.

Configuration Examples for Enabling Disk Mirroring

Enabling Disk Mirroring: Example

In the following example, disk mirroring is enabled on a router:

```
format disk1: partition

This operation will destroy all data on "disk1:" and partition device.
Continue? [confirm] y

Device partition disk1: is now formated and is available for use.

configure
  mirror location 0/0/cpu0 disk0:disk1:
  commit
```

show mirror Command Output: Example

```
RP/0/RSP0/CPU0:router(admin)# show mirror location all

Tue Dec  7 13:02:26.520 PST

Mirror Information for 0/RSP0/CPU0.
========================================================
 Mirroring Enabled
   Configured Primary:         disk0:
   Configured Secondary:       disk1:

 Current Mirroring State:      Redundant
   Current Physical Primary:   disk0:
   Current Physical Secondary: disk1:

 Mirroring Logical Device:     disk0:
 Mirroring Logical Device2:    disk1:

 Physical Device     State       Flags
--------------------------------------------------------
  disk0:           Available     Enabled
  disk1:           Available     Enabled
  compactflash:    Available
  (null)           Available
  disk0a:          Available
  disk1a:          Available
  compactflasha:   Not Present
  harddisk:        Available

Mirroring Rommon Variable
 BOOT_DEV_SEQ_CONF = disk0:;disk1:
 BOOT_DEV_SEQ_OPER = disk0:;disk1:
 MIRROR_ENABLE = Y
```

mirror verify Command Output: Example

```
RP/0/RSP0/CPU0:router# mirror verify

Mirror Verify Information for 0/0/CPU0.
========================================================
  Primary device and secondary device are fully synchronized.
```

Additional References

The following sections provide references related to disk mirroring configuration.

Related Documents

Related Topic	Document Title
Initial system bootup and configuration information for a router using the 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 command master list	*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*
Cisco IOS XR boot commands	*Boot Commands on the Cisco ASR 9000 Series Router* module of *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
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 16

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 32: 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

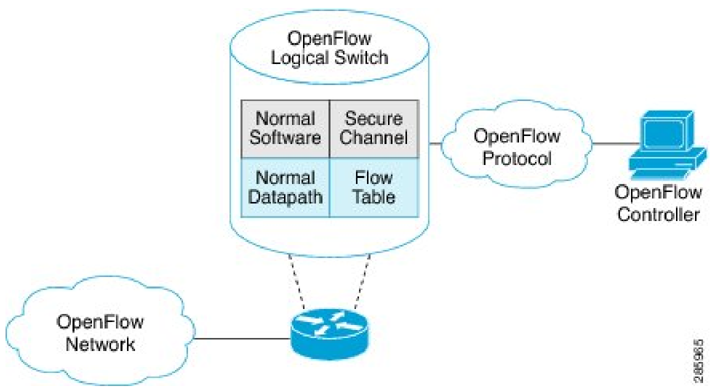
- enables customer-feature-independence

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 33: 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.

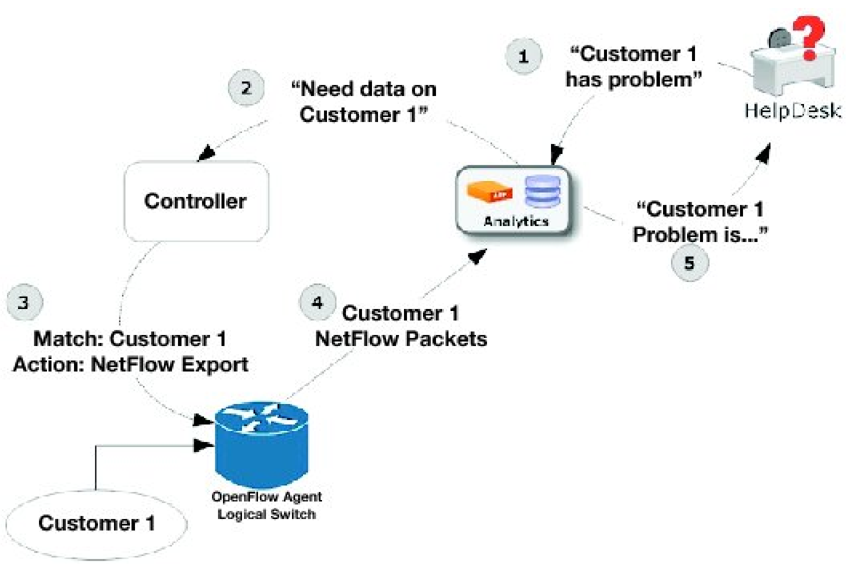
The following NetFlow maps must be configured:

- 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 8: 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.

	Command or Action	Purpose
		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 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:**	Enters the logical switch configuration mode. For L2_L3 switch, the pipeline number is 130.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-openflow)# switch 1 pipeline 130`	
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.

	Command or Action	Purpose
		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.

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 17

Configuring Call Home

This module describes the configuring of the Call Home feature.

Table 34: Feature History for Configuring Call Home

Release	Modification
Release 4.1.0	Call Home was introduced

This model contains the following topics:

About Call Home

Call Home provides an email and http/https based notification for critical system policies. A range of message formats are available for compatibility with pager services or XML-based automated parsing applications. You can use this feature to page a network support engineer, email a Network Operations Center, or use Cisco

Smart Call Home services to generate a case with the Technical Assistance Center. The Call Home feature can deliver alert messages containing information about diagnostics and environmental faults and events.

The Call Home feature can deliver alerts to multiple recipients, referred to as Call Home destination profiles. Each profile includes configurable message formats and content categories. A predefined destination is provided for sending alerts to the Cisco TAC, but you also can define your own destination profiles. When you configure Call Home to send messages, the appropriate CLI show command is executed and the command output is attached to the message. Call Home messages are delivered in the following formats:

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

Destination Profiles

A destination profile includes the following information:

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

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

The following predefined destination profiles are supported:

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

Call Home Alert Groups

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

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

Table 35: Alert Groups and Executed Commands

Alert Group	Description	Executed Commands
Environmental	Events related to power, fan, and environment-sensing elements such as temperature alarms.	**show environment** **show logging** **show inventory** **show environment trace** **show diag**
Inventory	Inventory status that is provided whenever a unit is cold booted, or when FRUs are inserted or removed. This alert is considered a noncritical event, and the information is used for status and entitlement.	**admin show platform** **admin show version** **admin show diag** **admin show inventory oid**
Syslog	Events generated by specific interesting syslog messages	**admin show version** **admin show logging** **admin show inventory**
Configuration	User-generated request for configuration or configuration change event.	• **show version** • **show running config all** • **show inventory** • **show configuration history last 30** • **show configuration commit changes last 1**
Snapshot	This alert group can be configured for periodic notifications	By default, this alert group has no commands to be run. You can add the required commands that need to be run.

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

Call Home Message Levels

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

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

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

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

Table 36: Severity and syslog Level Mapping

Call Home Level	Keyword	syslog Level	Description
9	Catastrophic	N/A	Network-wide catastrophic failure.
8	Disaster	N/A	Significant network impact.
7	Fatal	Emergency (0)	System is unusable.
6	Critical	Alert (1)	Critical conditions that indicate that immediate attention is needed.
5	Major	Critical (2)	Major conditions.
4	Minor	Error (3)	Minor conditions.
3	Warning	Warning (4)	Warning conditions.
2	Notification	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	Normal	Information (6)	Normal event signifying return to normal state.
0	Debugging	Debug (7)	Debugging messages.

Obtaining Smart Call Home

If you have a service contract directly with Cisco Systems, you can register your devices for the Smart Call Home service. Smart Call Home provides fast resolution of system problems by analyzing Call Home messages sent from your devices and providing background information and recommendations. For issues that can be identified as known, particularly GOLD diagnostics failures, Automatic Service Requests will be generated with the Cisco-TAC.

Smart Call Home offers the following features:

- Continuous device health monitoring and real-time diagnostic alerts.
- Analysis of Call Home messages from your device and, where appropriate, Automatic Service Request generation, routed to the appropriate TAC team, including detailed diagnostic information to speed problem resolution.

- Secure message transport directly from your device or through a downloadable Transport Gateway (TG) aggregation point. You can use a TG aggregation point in cases that require support for multiple devices or in cases where security requirements mandate that your devices may not be connected directly to the Internet.
- Web-based access to Call Home messages and recommendations, inventory and configuration information for all Call Home devices. Provides access to associated field notices, security advisories and end-of-life information.

You need the following items to register:

- The SMARTnet contract number for your device
- Your e-mail address
- Your Cisco.com ID

For more information about Smart Call Home, see the Smart Call Home page at this URL: https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome

Anonymous Reporting

Smart Call Home is a service capability included with many Cisco service contracts and is designed to assist customers resolve problems more quickly. If you decide not to use Smart Call Home, you can still enable Anonymous Reporting to allow Cisco to securely receive minimal error and health information from the device. If you enable Anonymous Reporting, your customer identity will remain anonymous, and no identifying information is sent.

When Call Home is configured for anonymous reporting, only , inventory, and test messages are sent to Cisco. No identifying information is sent.

Note When you enable Anonymous Reporting, you acknowledge your consent to transfer the specified data to Cisco or to vendors operating on behalf of Cisco (including countries outside the United States). Cisco maintains the privacy of all customers. For information about how Cisco treats personal information, see the Cisco Privacy Statement

Configuring Call Home

The tasks in this module describe how to configure the sending of Call Home messages. The following steps are involved:

1. Assign contact information.
2. Configure and enable one or more destination profiles.
3. Associate one or more alert groups to each profile.
4. Configure the email server options.
5. Enable Call Home.

> **Note** Before enabling Call-Home, you must configure the source interface for http over IPv6. However, for http over IPv4, Call-Home works without the source interface.
>
> In case of a dual-stack call-home configuration on the device, the IPv4 address is preferred over the IPv6 address. This may result in IPv6 resolution failure. Due to this limitation, the IPv6 device registration with the licensing server may only be done with a single mode, that is, IPv6 only configuration.
>
> Use the **http client source-interface ipv6** command to configure the source interface.

Configuring Contact Information

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

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **contact-email-addr** *email-address*
4. (Optional) **contract-id** *contract-id-string*
5. (Optional) **customer-id** *customer-id-string*
6. (Optional) **phone-number** *phone-number-string*
7. (Optional) **street-address** *street-address*
8. (Optional) **site-id** *site-id-string*
9. **commit**
10. **show call-home**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**call-home**
	Example:
	`RP/0/RSP0/CPU0:router(config)# call-home`
	`RP/0/RSP0/CPU0:router(config-call-home)#`
		Enters call home configuration mode.

Step 3	**contact-email-addr** *email-address*
	Example:
	`RP/0/RSP0/CPU0:router(config-call-home)#`
	`contact-email-addr`
	`user1@cisco.com`
		Configures the customer email address. Enter up to 200 characters in email address format with no spaces.

	Command or Action	Purpose
Step 4	(Optional) **contract-id** *contract-id-string* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)# contract-id Contract-identifier`	Configures the contract ID. Enter up to 64 characters. If you include spaces, you must enclose the entry in quotes ("").
Step 5	(Optional) **customer-id** *customer-id-string* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)# customer-id Customer1`	Configures the customer ID. Enter up to 64 characters. If you include spaces, you must enclose the entry in quotes ("").
Step 6	(Optional) **phone-number** *phone-number-string* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)# phone-number +405-123-4567`	Configures the customer phone number. The number must begin with a plus (+) prefix, and may contain only dashes (-) and numbers. Enter up to 16 characters.
Step 7	(Optional) **street-address** *street-address* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)# street-address "300 E. Tasman Dr. San Jose, CA 95134"`	Configures the customer street address where RMA equipment can be shipped. Enter up to 200 characters. If you include spaces, you must enclose the entry in quotes ("").
Step 8	(Optional) **site-id** *site-id-string* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)# site-id SJ-RouterRoom1`	Configures the site ID for the system. Enter up to 200 characters. If you include spaces, you must enclose the entry in quotes ("").
Step 9	**commit**	
Step 10	**show call-home** **Example:** `RP/0/RSP0/CPU0:router# show call-home`	Displays information about the system contacts.

Configuring and Activating Destination Profiles

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

SUMMARY STEPS

1. **configure**
2. **call-home**

3. **profile** *profile-name*
4. **destination address email** *email-address*
5. **destination message-size-limit** *max-size*
6. **destination preferred-msg-format** {**short-text** | **long-text** | **xml**}
7. **destination transport-method** [**email** | **hhtp**]
8. **active**
9. **commit**
10. **show call-home profile** {**all** | *profile-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**call-home**
	Example:
	`RP/0/RSP0/CPU0:router(config)# call-home`
	`RP/0/RSP0/CPU0:router(config-call-home)#`
		Enters call home configuration mode.

Step 3	**profile** *profile-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-call-home)# profile my_profile`
	`RP/0/RSP0/CPU0:router(config-call-home-profile)#`
		Enters call home profile configuration mode to configure a new or existing profile.

Step 4	**destination address email** *email-address* **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# destination address email support_me@cisco.com`	Configures an email address to which Call Home messages are sent for this profile.
Step 5	**destination message-size-limit** *max-size* **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# destination message-size-limit 1000`	Configures the maximum size of Call Home messages for this profile. Values can be between 50 and 3145728 characters.
Step 6	**destination preferred-msg-format** {**short-text** | **long-text** | **xml**} **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# destination preferred-msg-format xml`	Configures the message format for this profile. The default is xml.

	Command or Action	Purpose
Step 7	**destination transport-method** [**email** | **hhtp**] **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# destination transport-method email`	Configures the transport method for this profile.
Step 8	**active** **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# active`	Activates the destination profile. **Note** At least one destination profile must be active for Call Home messages to be sent.
Step 9	**commit**	
Step 10	**show call-home profile** {**all** | *profile-name*} **Example:** `RP/0/RSP0/CPU0:router# show call-home profile all`	Displays information about the destination profile.

Associating an Alert Group with a Destination Profile

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

Before you begin

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

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **profile** *profile-name*
4. **subscribe-to-alert-group environment** [**severity** *severity-level*
5. **subscribe-to-alert-group inventory** [**periodic** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hh:mm*
6. **subscribe-to-alert-group syslog severity** *severity-level* **pattern** *string*
7. **subscribe-to-alert-group snapshot severity** *severity-level* **pattern** *string*
8. **subscribe-to-alert-group configuration severity** *severity-level* **pattern** *string*
9. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose

Step 2	**call-home**
	Example:
	`RP/0/RSP0/CPU0:router(config)# call-home`
	`RP/0/RSP0/CPU0:router(config-call-home)#`
		Enters call home configuration mode.

Step 3	**profile** *profile-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-call-home)# profile my_profile`
	`RP/0/RSP0/CPU0:router(config-call-home-profile)#`
		Enters call home profile configuration mode to configure a new or existing profile.

Step 4	**subscribe-to-alert-group environment** [**severity** *severity-level* **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group environment severity major`	Configures a destination profile to receive messages for the environment alert group. Alerts with a severity the same or greater than the specified severity level are sent. • **catastrophic**—Includes network-wide catastrophic events in the alert. This is the highest severity. • **critical**—Includes events requiring immediate attention (system log level 1). • **disaster**—Includes events with significant network impact. • **fatal**—Includes events where the system is unusable (system log level 0). • **major**—Includes events classified as major conditions (system log level 2). • **minor**—Includes events classified as minor conditions (system log level 3) • **normal**—Specifies the normal state and includes events classified as informational (system log level 6). This is the default. • **notification**—Includes events informational message events (system log level 5). • **warning**—Includes events classified as warning conditions (system log level 4).
Step 5	**subscribe-to-alert-group inventory** [**periodic** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hh:mm* **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group inventory periodic monthly 1 10:00`	Configures a destination profile to receive messages for the inventory alert group. Either alerts are sent periodically, or any non-normal event triggers an alert.

	Command or Action	Purpose
Step 6	**subscribe-to-alert-group syslog severity** *severity-level* **pattern** *string* **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group syslog severity major pattern`	Configures a destination profile to receive messages for the syslog alert group. Alerts with a severity the same or greater than the specified severity level are sent. • **catastrophic**—Includes network-wide catastrophic events in the alert. This is the highest severity. • **critical**—Includes events requiring immediate attention (system log level 1). • **disaster**—Includes events with significant network impact. • **fatal**—Includes events where the system is unusable (system log level 0). • **major**—Includes events classified as major conditions (system log level 2). • **minor**—Includes events classified as minor conditions (system log level 3) • **normal**—Specifies the normal state and includes events classified as informational (system log level 6). This is the default. • **notification**—Includes events informational message events (system log level 5). • **warning**—Includes events classified as warning conditions (system log level 4). You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").
Step 7	**subscribe-to-alert-group snapshot severity** *severity-level* **pattern** *string* **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group snapshot severity major pattern`	Configures a destination profile to receive messages for the snapshot alert group. Alerts with a severity the same or greater than the specified severity level are sent. You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").
Step 8	**subscribe-to-alert-group configuration severity** *severity-level* **pattern** *string* **Example:** `RP/0/RSP0/CPU0:router(config-call-home-profile)# subscribe-to-alert-group configuration severity major pattern`	Configures a destination profile to receive messages for the configuration alert group. Alerts with a severity the same or greater than the specified severity level are sent. You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes ("").
Step 9	**commit**	

What to do next

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

Configuring Email

Call Home messages are sent via email. You must configure your email server before Call Home messages can be sent.

SUMMARY STEPS

1. **configure**
2. **call-home**
3. (Optional) **sender from** *email-address*
4. (Optional) **sender reply-to** *email-address*
5. **mail-server** *address* **priority** *priority*
6. **rate-limit** *events-count*
7. **commit**
8. **show call-home mail-server status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**call-home**
	Example:
	`RP/0/RSP0/CPU0:router(config)# call-home`
	`RP/0/RSP0/CPU0:router(config-call-home)#`
		Enters call home configuration mode.

Step 3	(Optional) **sender from** *email-address* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)# sender from my_email@cisco.com`	Specifies the email message "from" address.
Step 4	(Optional) **sender reply-to** *email-address* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)# sender reply-to my_email@cisco.com`	Specifies the email message "reply-to" address.
Step 5	Required: **mail-server** *address* **priority** *priority* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)#`	Specifies the mail server to use to send Call Home messages. You can specify an IP address or mail server name. You can specify up to five mail servers to use. The server with the lower priority is tried first.

	Command or Action	Purpose
	`mail-server 198.51.100.10 priority 1`	
Step 6	Required: **rate-limit** *events-count* **Example:** `RP/0/RSP0/CPU0:router(config-call-home)# rate-limit 4`	Specifies the maximum trigger rate per minute. The default is five events per minute and the maximum is also five.
Step 7	**commit**	
Step 8	**show call-home mail-server status** **Example:** `RP/0/RSP0/CPU0:router# show call-home mail-server status`	Displays the status of the specified mail server.

Enabling Call Home

By default the sending of Call Home messages is disabled. You must perform this task to enable the sending of Call Home messages.

Before you begin

Before enabling the sending of Call Home messages, you should complete the configuration tasks described in this module. Specifically, you must have enabled a destination profile for any Call Home messages to be sent.

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **service active**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**call-home**
	Example:
	`RP/0/RSP0/CPU0:router(config)# call-home`
	`RP/0/RSP0/CPU0:router(config-call-home)#`
		Enters call home configuration mode.

Step 3	**service active** **Example:**	Enables the sending of Call Home messages.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-call-home)# service active`	
Step 4	**commit**	

Configuring Smart Call Home (single command)

SUMMARY STEPS

1. **configure**
2. **call-home reporting** { **anonymous** \| **contact-email** *email-address* } [**http-proxy** { *address* } **port** *port-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home reporting** { **anonymous** | **contact-email** *email-address* } [**http-proxy** { *address* } **port** *port-number*] **Example:** `RP/0/RSP0/CPU0:router (config) #` **`call-home reporting contact-email`** ***`email@company.com`***	Enables all call home basic configurations using a single command.

Configuring Call Home Data Privacy

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **data-privacy** { **level** { **normal** \| **high** } \| **hostname** }

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home** **Example:** `RP/0/RSP0/CPU0:router(config) #` **`call-home`**	Enters the call home configuration submode.

	Command or Action	Purpose
Step 3	**data-privacy** { **level** { **normal** | **high** } | **hostname** } **Example:** `RP/0/RSP0/CPU0:router(config-call-home) #` **`data-privacy level high`**	Scrubs data from call-home message to protect the privacy of the user. The default data-privacy level is normal. • **normal** - scrubs all normal level commands , such as , password/ username/ ip/ destination. • **high** - scrubs all normal level commands plus the IP domain name and IP address commands. • **hostname** - scrubbing the hostname from call-home messages may cause Smart Call Home processing failure. **Note** Enabling the data-privacy command can affect CPU utilization when scrubbing a large amount of data.

Configuring Syslog Throttling

This task is used to enable or disable Call Home syslog message throttling and avoid sending repetitive Call Home syslog messages.

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **syslog-throttling**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home** **Example:** `RP/0/RSP0/CPU0:router (config) #` **`call-home`**	Enters call home configuration submode.
Step 3	**syslog-throttling** **Example:** `RP/0/RSP0/CPU0:router (config-call-home) #` **`syslog-throttling`**	Enables or disables Call Home syslog message throttling and avoids sending repetitive Call Home syslog messages. By default, syslog message throttling is enabled.

Enabling AAA Authorization

This task is used to enable AAA authorization for Call Home messages.

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **aaa-authorization** [**username** *username*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home** **Example:** `RP/0/RSP0/CPU0:router (config) # call-home`	Enters Call Home configuration mode.
Step 3	**aaa-authorization** [**username** *username*] **Example:** `RP/0/RSP0/CPU0:router (config-call-home) # aaa-authorization username u1`	Enables AAA authorization. Specifies the username for authorization.

Sending Call Home Alert group Messages Manually

This task is used to manually trigger Call Home alert group messages.

You can use the **call-home send** command to manually send a specific alert group message. Guidelines for the CLI options of the command:

- Only the snapshot, configuration, and inventory alert groups can be sent manually. Syslog alert groups cannot be sent manually.
- When you manually trigger a snapshot, configuration, or inventory alert group message and you specify a destination profile name, a message is sent to the destination profile regardless of the profile's active status, subscription status, or severity setting.
- When you manually trigger a snapshot, configuration, or inventory alert group message and do not specify a destination profile name, a message is sent to all active profiles that have either a normal or periodic subscription to the specified alert group.

SUMMARY STEPS

1. **call-home send alert-group snapshot** [**profile** *name*]
2. **call-home send alert-group configuration** [**profile** *name*]
3. **call-home send alert-group inventory** [**profile** *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**call-home send alert-group snapshot** [**profile** *name*] **Example:** `RP/0/RSP0/CPU0:router # `**`call-home send alert-group snapshot profile p1`**	Sends a snapshot alert group message to one destination profile if specified or to all subscribed destination profiles.
Step 2	**call-home send alert-group configuration** [**profile** *name*] **Example:** `RP/0/RSP0/CPU0:router # `**`call-home send alert-group configuration profile p1`**	Sends a configuration alert group message to one destination profile if specified or to all subscribed destination profiles.
Step 3	**call-home send alert-group inventory** [**profile** *name*] **Example:** `RP/0/RSP0/CPU0:router # `**`call-home send alert-group inventory profile p1`**	Sends an inventory alert group message to one destination profile if specified or to all subscribed destination profiles.

Manually sending command output message for a Command List

You can use the **call-home send** command to execute a command or a list of commands and send the command output through HTTP or email protocol.

Guidelines when sending the output of a command:

- The specified command or list of commands can be any run command, including commands for all modules. The command must be contained in quotes (“”).
- If the email option is selected using the “email” keyword and an email address is specified, the command output is sent to that address.
- If neither the email nor the HTTP option is specified, the output is sent in long-text format with the specified service request number to the Cisco TAC (attach@cisco.com).
- If neither the “email” nor the “http” keyword is specified, the service request number is required for both long-text and XML message formats and is provided in the subject line of the email.
- If the HTTP option is specified, the CiscoTAC-1 profile destination HTTP or HTTPS URL is used as the destination. The destination email address can be specified so that Smart Call Home can forward the message to the email address. The user must specify either the destination email address or an SR number but they can also specify both.

This task enables you to execute command and send the command output.

SUMMARY STEPS

1. **call-home send** { *cli command* | *cli list* } [**email** *email* **msg-format** { **long-text** | **xml** } | **http** { **destination-email-address** *email* }] [**tac-request** *SR#*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**call-home send** { *cli command* | *cli list* } [**email** *email* **msg-format** { **long-text** | **xml** } | **http** { **destination-email-address** *email* }] [**tac-request** *SR#*] **Example:** `RP/0/RSP0/CPU0:router #  call-home send "show version;show running-config;show inventory" email support@example.com msg-format xml`	Executes the CLI or CLI list and sends output via email or HTTP. • { cli command | cli list }—Specifies the command or list of commands (separated by ';'). It can be any run command, including commands for all modules. The commands must be contained in quotes (""). • **email** email **msg-format { long-text | xml }**—If the email option is selected, the command output will be sent to the specified email address in long-text or XML format with the service request number in the subject. The email address, the service request number, or both must be specified. The service request number is required if the email address is not specified (default is attach@cisco.com for long-text format and callhome@cisco.com for XML format). • **http { destination-email-address** email }—If the http option is selected, the command output will be sent to Smart Call Home backend server (URL specified in the CiscoTAC-1 profile) in XML format. destination-email-address email can be specified so that the backend server can forward the message to the email address. The email address, the service request number, or both must be specified. • **tac-service-request** SR# —Specifies the service request number. The service request number is required if the email address is not specified.

Configuring a HTTP Proxy Server

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

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **http-proxy** *proxy-server-name* **port** *port-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home** **Example:** `RP/0/RSP0/CPU0:router (config) # call-home`	Enters Call Home configuration mode.
Step 3	**http-proxy** *proxy-server-name* **port** *port-number* **Example:** `RP/0/RSP0/CPU0:router (config) # http-proxy p1 port 100`	Configures the port for the specified HTTP proxy server. Range is 1 to 65535.

Configuring Snapshot alert group

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **alert-group-configuration snapshot**
4. **add-command** *"command string"*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home** **Example:** `RP/0/RSP0/CPU0:router (config) # call-home`	Enters Call Home configuration mode.
Step 3	**alert-group-configuration snapshot** **Example:** `RP/0/RSP0/CPU0:router (config-call-home) # alert-group-configuration snapshot`	Enters snapshot configuration mode.
Step 4	**add-command** *"command string"* **Example:** `RP/0/RSP0/CPU0:router (config-call-home-snapshot) # add-command "show ver"`	Adds the command to the snapshot alert group.

Configuring Anonymous Reporting

This task enables the user to configure an anonymous mode profile.

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **profile** *name*
4. **anonymous-reporting-only**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home** **Example:** `RP/0/RSP0/CPU0:router (config) # call-home`	Enters Call Home configuration mode.
Step 3	**profile** *name* **Example:** `RP/0/RSP0/CPU0:router (config-call-home) # profile ciscotac`	Enters the profile configuration mode.
Step 4	**anonymous-reporting-only** **Example:** `RP/0/RSP0/CPU0:router (config-call-home-profile) # anonymous-reporting-only`	Enters anonymous mode. When **anonymous-reporting-only** is set, only inventory and test messages are sent.

What to do next

-

Configuring Call Home to use VRF

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home** **Example:** `RP/0/RSP0/CPU0:router (config) # call-home`	Enters Call Home configuration mode.
Step 3	**vrf** *vrf-name* **Example:** `RP/0/RSP0/CPU0:router (config) # vrf v1`	Configures call home for the specified VRF. VRF works only for the http transport method. It does not work for the email transport method.

Configuring Source Interface

This task enables the user to configure a source interface.

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **source-interface** *type interface-path-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**call-home** **Example:** `RP/0/RSP0/CPU0:router (config) # call-home`	Enters Call Home configuration mode.
Step 3	**source-interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router (config) # source-interface tengige 10.1.1.1`	Configures the source interface. **Note** Source-interface supports email and HTTP messages.

CHAPTER 18

Configuring Data Collection Manager

This module describes the configuring of the Data Collection Manager feature.

Table 37: 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 315
	`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 317	`bulkstat instance if-mib type snmp` ` wildcard`

Step4: Configuring **Data Group if-group**
For detailed procedure:
Configuring a Bulkstat Data Group, on page 318
	`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 320
	`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 315
	`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 317	`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 316
	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 318
	`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 320
	`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 315	`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 318
	`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 320
	`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 19

Software Entitlement

Cisco IOS XR software contains all the supported features for a given release. Before the introduction of software entitlement on Cisco IOS XR software, you could freely activate all available software packages on your network devices and could enable all the bundled features. Software entitlement has been introduced so you pay only for the features that you need today, but can upgrade when necessary while keeping your investment safe. Licensing enables you to purchase individual software features and upgrade hardware capacity in a safe and reliable way.

The licensing methods supported on Cisco IOS XR software are:

- Smart Licensing
- Default (traditional) Licensing

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*.

This model contains the following topics:

What Is Software Entitlement?

Software entitlement is a system that consists of a license manager on a Cisco IOS XR device that manages licenses for various software and hardware features. The license manager parses and authenticates a license before accepting it. The software features on the router use the license manager APIs to check out and release licenses. Licenses are stored in persistent storage on the router.

Core routing features are available for use without any license. The following features can be enabled on your router using licenses:

Layer 3 VPN

Layer 3 (virtual private network) VPN can be configured only if there is an available Layer 3 VPN license for the line card slot on which the feature is being configured. If the advanced IP license is enabled, 4096 Layer 3 VPN routing and forwarding instances (VRFs) can be configured on a line card. If the infrastructure VRF license is enabled, eight Layer 3 VRFs can be configured on the line card.

To activate the Infrastructure VRF license, you need to configure two interfaces or sub-interfaces in separate VRFs, with at least one physical interface in each of the VRFs.

The key is to have multiple (two or more) user-defined VRFs configured in at least one slot and at least one physical interface in each user-defined VRF; and repeated for each slot.

In a non-consumption model line card, configuring a physical interface in multiple VRFs would consume a L3VPN license. However, configuring other virtual interfaces (such as management, or BVI interfaces) in multiple VRFs would'nt consume L3VPN license. Configuring VRFs under bundle main and sub-interfaces does consume L3VPN license.

See the following modules in *MPLS Configuration Guide for Cisco ASR 9000 Series Routers* for information about Layer 3 VPN configurations:

- *Implementing MPLS Layer 3 VPNs on the Cisco ASR 9000 Series Router*
- *Implementing Virtual Private LAN Services on the Cisco ASR 9000 Series Router*

G.709

If a G.709 license is available, G.709 can be enabled on 10-Gigabit Ethernet interfaces on the following line cards:

- 2-port 10 Gigabit Ethernet / 20-port Gigabit Ethernet line card
- 8-port 10 Gigabit Ethernet line card
- 24-port 10 Gigabit Ethernet line card
- 36-port 10 Gigabit Ethernet line card
- 4-port 10 Gigabit Ethernet modular port adapter
- 2-port 10 Gigabit Ethernet modular port adapter

Refer to the *Configuring Dense Wavelength Division Multiplexing Controllers on the Cisco ASR 9000 Series Router* module in *Interface and Hardware Component Configuration Guide for Cisco ASR 9000 Series Routers*.

Video Monitoring

Video monitoring can be enabled for the Cisco ASR 9000 chassis by using a video monitoring license.

Satellite Network Virtualization (nV)

The Satellite nV license entitles satellite devices to connect to the Cisco ASR 9000 chassis. Satellite licenses are chassis licenses, and can provide the ability for one, five or 20 satellites to connect to a Cisco ASR 9000 host remotely.

Note Smart Licensing is supported on a cluster set-up. There are two A9K-NV-CLUSTR-LIC licenses required/requested, one for each of the chassis.

Implementing Smart Licensing

Information About Smart Licensing

Smart Licensing is a cloud-based, software license management solution that enables you to automate time-consuming, manual licensing tasks. The solution allows you to easily track the status of your license and software usage trends.

Smart Licensing helps simplify three core functions:

- **Purchasing:** The software that you have installed in your network can automatically self-register themselves, without Product Activation Keys (PAKs).
- **Management:** You can automatically track activations against your license entitlements. Additionally, there is no need to install the license file on every node. You can create license pools (logical grouping of licenses) to reflect your organization structure. Smart Licensing offers you Cisco Smart Software Manager, a centralized portal that enables you to manage all your Cisco software licenses from one centralized website. Cisco Smart Software Manager Overview, on page 355 provides details.
- **Reporting:** Through the portal, Smart Licensing offers an integrated view of the licenses you have purchased and what has been actually deployed in your network. You can use this data to make better purchase decisions, based on your consumption.

Smart Versus Traditional Licensing

Traditional (node locked) licencing	Smart (dynamic) licencing
You must procure the license and manually install it on the device.	Your device initiates a call home and requests the licenses it needs. *Configuring Call Home on the Cisco ASR 9000 Series Router* describes the Smart Call Home feature.
Node-locked licences - license is associated with a specific device.	Pooled licences - licences are company account-specific, and can be used with any compatible device in your company. You can activate or deactivate different types of licenses on the device without actually installing a license file on the device.
No common install base location to view licenses purchased or software usage trends	Licenses are stored securely on Cisco servers accessible 24x7x365.
No easy means to transfer licenses from one device to another.	Licenses can be moved between product instances without a license transfer. This greatly simplifies the reassignment of a software license as part of the Return Material Authorization (RMA) process.
Limited visibility into all software licenses being used in the network. Licenses are tracked only on per node basis.	Complete view of all Smart Software Licenses used in the network using a consolidated usage report of software licenses and devices in one easy-to-use portal.

Smart Licensing for Cisco IOS XR 64 bit

Table 38: Feature History Table

Feature Name	Release Information	Feature Description
Support for Smart Licensing on ASR 9902 Routers, 0.8T PEC, and ASR 9000 5th generation 400G Line Cards.	Release 7.4.1	Smart Licensing support is now extended to the following routers and line cards: • ASR-9902 • A9K-8HG-FLEX-SE/TR • A9K-4HG-FLEX-SE/-TR ine cards • A99-4HG-FLEX-SE/TR line cards

Cisco IOS XR 64 bit does not support Traditional Licensing. Only Smart Licensing is supported in Cisco IOS XR 64 bit. Smart Licensing is enabled by default on Cisco IOS XR 64 bit. Here are few things to consider for Smart Licensing in Cisco IOS XR 64 bit:

- Smart Licensing cannot be disabled.
- All the Smart Licensing CLIs are executed from EXEC mode.
- EVAL Period is not supported. Hence licenses are consumed only after registeration.

This table shows license consumption logic for non-Consumption Model (CM) line cards:

Table 39: Feature wise requirement for Non-CM Line Cards

License	Eligibility Criteria
AIP	If number of VRFs are greater than 8, AIP license is consumed.
IVRF	If number of VRFs are less than or equal to 8, IVRF license is consumed. If customer has less than or equal to 8 VRFs, who purchased AIP license in their smart account, please contact your Cisco account representative if you see issues like insufficient IVRF licenses.

This table lists supported licenses for non-Consumption Model (CM) line cards:

Table 40: Non-CM Line Cards Software Licenses for Cisco IOS XR 64-bit

Non-CM Line Cards Software License PID	Description
A9K-800G-IVRF	ASR 9000 8-port 100 GE Infrastructure VRF Right to Use License

Non-CM Line Cards Software License PID	Description
A9K-400G-IVRF	ASR 9000 4-port 100 GE Infrastructure VRF license Right to Use License
A99-1200G-IVRF	ASR 9900 Infrastructure VRF license Right to Use License
A9K-800G-AIP-SE	ASR 9000 8-port 100 GE Advance IP Service Edge Right to Use License
A9K-800G-AIP-TR	ASR 9000 8-port 100 GE Advance IP Transport Optimised Right to Use License
A9K-400G-AIP-SE	ASR 9000 4-port 100 GE Advance IP Service Edge Right to Use License
A9K-400G-AIP-TR	ASR 9000 4-port 100 GE Advance IP Transport Optimised Right to Use License
A99-1200G-AIP	ASR 9900 12port 100GE Advanced IP Right to Use License
A9K-800G-OPT-LIC	ASR 9000 8-port 100 GE Advance Optical Right to Use License
A9K-400G-OPT-LIC	ASR 9000 4-port 100 GE Advance Optical Right to Use License
A99-1200G-ADVRTNG	ASR 9900 12port 100GE Advanced Routng Right to Use License
S-A9K-9901-AIP-LC	ASR 9000 Smart License L3 VPN for NON PAYG 9901 System
S-A9K-9901-VRF-LC	ASR 9000 Smart License I-VRF for NON PAYG 9901 System
S-A9K-9901-120AIP	ASR 9000 Smart License L3 VPN for 120G PAYG 9901 System
S-A9K-9901-256AIP	ASR 9000 Smart License L3 VPN for 256G PAYG 9901 System
S-A9K-BNG-ADV-8K	ASR 9000 Smart License for BNG license for Advance Features
S-A9K-BNG-LIC-8K	ASR 9000 BNG License Unit for 8000 subscribers
A9K-24P-80GRTU-SE	ASR 9000 80G Upgrade license for 24-port 10G/1G dual rate Service Edge
A9K-24P-80GRTU-TR	ASR 9000 80G Upgrade license for 24-port 10G/1G dual rate Transport Optimized

Non-CM Line Cards Software License PID	Description
A9K-48P1GE-AIP-SE	ASR 9000 Advance IP Service Edge License for full scale VRFs for 48-port 1G mode
A9K-48P1GE-AIP-TR	ASR 9000 Advance IP Transport Optimized License for full scale VRFs for 48-port 1G mode
A9K-48P10G-SE-UPG	ASR 9000 48-port 1G to 10G Service Edge Upgrade License

Note Currently, A9K-48P1GE-AIP-TR license is not supported on Cisco IOS XR 32 bit software.

Consumption Model licenses are of of two types: Foundation and Premium. This table lists supported licenses for CM line cards:

Foundation Licenses

Foundation	License PID	Eligibility Criteria	License PID
L2VPN	S-A9K-L2-10G/1G/100G	per 10G	Check the output of the following commands: • **show l2vpn xconnect** • **show l2vpn bridge-domain** Each main-interface in the output qualifies for a license.
L3VPN	S-A9K-L3-10G/1G/100G	per 10G	Check the output of the following commands: • show ipv4 vrf all interface brief | ex default • show ipv6 vrf all interface brief | ex default First, remove all internal VRFs form the output (VRF names starting with "**"). Each remaining main-interface in the output qualifies for a license.

IP/MPLS	S-A9K-IP-10G/1G/100G	per 10G	**1.** Check if main-interface qualifies for L2VPN Foundation License. **2.** Check if main-interface qualifies for L3VPN Foundation License. **3.** Check if main-interface is in "no shutdown" state. **4.** If any of 1, 2 and 3 is met, main-interface qualifies for a license.

Configuration Threshold for Premium Licenses

Configuration	Threshold Value
L2 Subinterfaces	4,096
Bridge Domain	1,024
MAC scale	32,000
L3 Routes (IPv4/IPv6 global)	132,000
MPLS TE (Head+Tail)	1,024
VRF scale	8
L3 Subinterfaces	1,024

This is the list of show commands available for Smart Licensing in Cisco IOS XR 64 bit :

- show license all—shows all information regarding Smart license
- show license platform—shows platform-specific licensing information (cisco-support)
- show license status—shows smart licensing status information
- show license summary—shows smart licensing summary
- show license techsupport—shows smart licensing tech support information
- show license trace—shows tracing for smart licensing code (cisco-support)
- show license udi—shows smart licensing UDI information
- show license usage—shows smart licensing usage information
- show license platform detail—shows smart license detail information (cisco-support)

- show license platform summary—shows smart license summary (cisco-support)
- show license platform trace—shows platform specific licensing trace information (cisco-support)

Create a Cisco Smart Account

Cisco Smart Account is an account where all products enabled for Smart Licensing are deposited. Cisco Smart Account allows you to manage and activate your licenses to devices, monitor license use, and track Cisco license purchases. Through transparent access, you have a real-time view into your Smart Licensing products. IT administrators can manage licenses and account users within your organization's Smart Account through the Smart Software Manager.

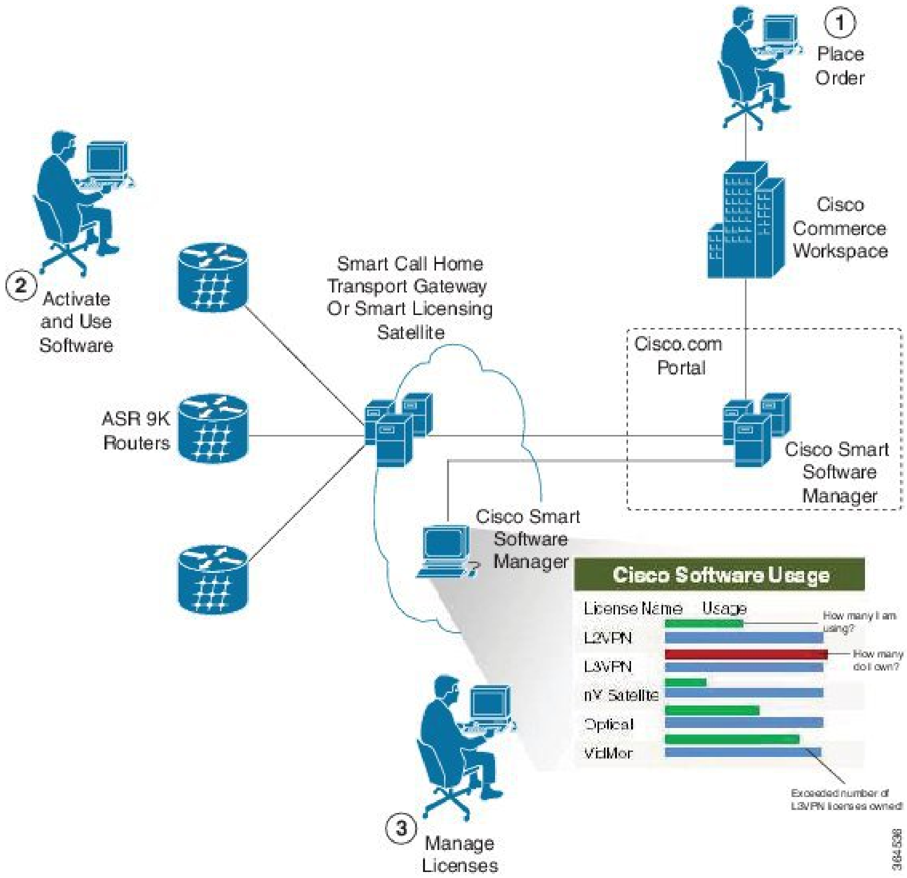
You can create your Cisco Smart Account at this webpage: https://webapps.cisco.com/software/company/smartaccounts/home#accountcreation-account/request.

For information on how to create a Cisco Smart Account, see: http://www.cisco.com/c/en/us/products/collateral/software/one-software/solution-overview-c22-733273.html.

Smart Licensing Working

Smart Licensing involves the three steps shown in the illustration below, that depicts the working model of the Smart Licensing.

Figure 9: Smart Licensing - Example



- **Setting up Smart Licensing:** You can place the order for Smart Licensing, to manage licenses on Cisco.com portal. You agree to the terms and conditions governing the use and access of Smart Licensing in the Smart Software Manager portal at http://www.cisco.com/c/en/us/products/collateral/software/one-software/solution-overview-c22-733273.html.
- **Enabling and Use Smart Licensing:** Follow the steps to enable Smart Licensing. *Smart Licencing Workflow* provides an illustration.

  After you enable Smart Licensing, you can use either of the following options to communicate:

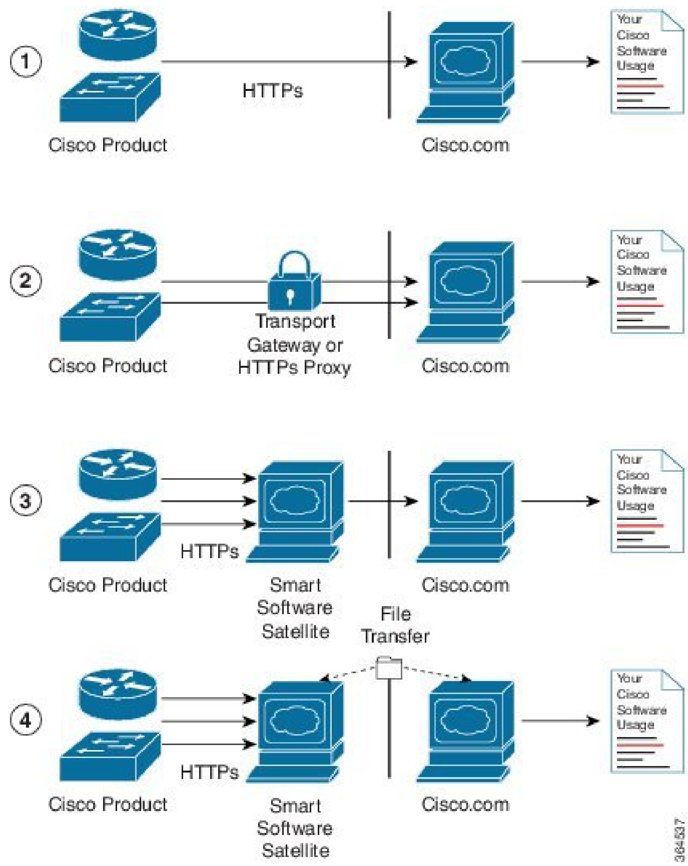
  - **Smart Call Home:** The Smart Call Home feature is automatically configured after the Smart Licensing is enabled. Smart Call Home is used by Smart Licensing as a medium for communication with the Cisco license service. Call Home feature allows Cisco products to periodically call-home and perform an audit and reconciliation of your software usage information. This information helps Cisco efficiently track your install base, keep them up and running, and more effectively pursue service and support contract renewals, without much intervention from your end. For more information on Smart Call Home feature, see http://www.cisco.com/c/dam/en/us/td/docs/switches/lan/smart_call_home/SCH_Deployment_Guide.pdf.
  - **Smart Licensing Satellite:** The Smart licensing satellite option provides an on-premises collector that can be used to consolidate and manage Smart license usage, as well facilitate communications back to Cisco License Service at http://www.cisco.com.

- **Manage and Report Licenses:** You can manage and view reports about your overall software usage in the Smart Software Manager portal. Compliance reporting, on page 356 describes the types of Smart Licensing reports.

Deployment Options for Smart Licensing

The following illustration shows the various options available for deploying Smart Licensing:

Figure 10: Smart Licensing Deployment Options



1. **Direct cloud access:** In direct cloud access deployment method, Cisco products send usage information directly over the internet to Cisco.com (Cisco license service); no additional components are needed for deployment.

2. **Direct cloud access through an HTTPs proxy:** In direct cloud access through an HTTPs proxy deployment method, Cisco products send usage information over the internet through a proxy server - either a Smart Call Home Transport Gateway or off-the-shelf Proxy (such as Apache) to Cisco License Service on http://www.cisco.com.

3. **Mediated access through an on-premises collector-connected:** In mediated access through an on-premises collector-connected deployment method, Cisco products send usage information to a locally-connected collector, which acts as a local license authority. Periodically, the information is exchanged to keep the databases in synchronization.

4. **Mediated access through an on-premises collector-disconnected:** In the mediated access through an on-premises collector-disconnected deployment method, Cisco products send usage information to a local disconnected collector, which acts as a local license authority. Exchange of human-readable information is performed occasionally (maybe once a month) to keep the databases in synchronization.

Options **1** and **2** provide an easy deployment option, and options **3** and **4** provide a secure environment deployment option. Smart Software Satellite provides support for options **3** and **4**.

The communication between Cisco products and Cisco license service is facilitated by the Smart Call Home software. For information on Smart Call Home, see About Call Home, on page 285

Configure Licenses Using Smart Licensing

Pre-requisites for Configuring Smart Licensing

The following pre-requistites must be met on the Cisco Smart Software Manager to configure Smart Licensing on your device:

- You must set up a Cisco Smart Account.
- You must set up Virtual Account or accounts. For more information, see the Virtual Accounts section in the Smart Software Manager Help
- Create user roles in the **Users** tab in the **Manage Smart Account** page. Provide the appropriate user access rights.
- Accept the Smart Software Licensing Agreement on Cisco Smart Software Manager to register your router.
- Have a layer 3 connection set up on your router.
- Configure a valid DNS and proper time on the router to connect CSSM or CSSM On-Prem.

Setting up the Router for Smart Licensing

Table 41: Three-step Roadmap to Set up the Router for Smart Licensing

Activity	Communication Connection Options		
Step 1—Configure Communications	See the *Configuring a Direct Cloud Connection* section.	See the *Configuring a Connection through a HTTP Proxy* section.	See the *Connecting to CSSM On-Premise* section.
Step 2—Register and Activate	See the *Registering and Activating your Router* section.		
Step 3—Verify the Configuration	See the *Verifying your Smart Licensing Configuration* section.		

Configuring a Communications Connection Between the Router and Cisco Smart Software Manager

Configuring a Direct Cloud Connection

In this deployment option, the **configure call-home profile** is configured by default. Use the **show call-home profile all** command to check the profile status.

Call Home service provides email-based and web-based notification of critical system events to Cisco Smart Software Manager.

To configure and enable Call Home service:

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **service active**
4. **contact-email-addr** *email-address*
5. **profile CiscoTAC-1**
6. **destination transport-method http**
7. **destination address http** *url*
8. **active**
9. **no destination transport-method email**
10. **commit**
11. **exit**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `Router# configureteral`	Enters global configuration mode.
Step 2	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.
Step 3	**service active** **Example:** `Router(config-call-home)# service active`	Activates Call Home service.
Step 4	**contact-email-addr** *email-address* **Example:** `Router(config-call-home)# contact-email-addr username@example.com`	Assigns the provided email address. You can enter up to 200 characters in email address format. **Note** Spaces are not allowed in the email address.

	Command or Action	Purpose
Step 5	**profile CiscoTAC-1** **Example:** `Router(config-call-home)# profile CiscoTAC-1`	Enables the CiscoTAC-1 profile to be used with the Call Home service. By default, the CiscoTAC-1 profile is disabled.
Step 6	**destination transport-method http** **Example:** `Router(config-call-home-profile)# destination transport-method http`	Enables the Call Home service through an HTTP connection.
Step 7	**destination address http** *url* **Example:** `Router(config-call-home-profile)# destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService`	Connects the router to the Cisco Smart Software Manager.
Step 8	**active** **Example:** `Router(config-call-home-profile)# active`	Enables the destination profile.
Step 9	**no destination transport-method email** **Example:** `Router(config-call-home-profile)# no destination transport-method email`	Disables the email option for the Call Home service.
Step 10	**commit** **Example:** `Router(config-call-home-profile)# commit`	Commits the configuration.
Step 11	**exit** **Example:** `Router(config-call-home-profile)# exit`	Exits the Call Home destination profile configuration mode and returns to the Call Home configuration mode.

Step 12	**exit**
	Example:
	`Router(config-call-home)# exit`
	`Router(config)#`
		Exits the Call Home configuration mode and returns to the global configuration mode.

Configuring a Connection Through an HTTP Proxy

The Call Home service can be configured through an HTTPs proxy server.

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **service active**
4. **contact-email-address** *email-address*

5. **http-proxy** *proxy-address* **port** *port-number*
6. **profile CiscoTAC-1**
7. **no destination transport-method email**
8. **exit**
9. **profile** *profile-name*
10. **reporting smart-licensing-data**
11. **destination transport-method http**
12. **destination address http** *url*
13. **active**
14. **exit**
15. **exit**
16. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `Router# configure`	Enters global configuration mode.
Step 2	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.
Step 3	**service active** **Example:** `Router(config-call-home)# service active`	Enables the Call Home feature.
Step 4	**contact-email-address** *email-address* **Example:** `Router(config-call-home)# contact-email-addr sch-smart-licensing@cisco.com`	Configures the default email address.
Step 5	**http-proxy** *proxy-address* **port** *port-number* **Example:** `Router(config-call-home)# http-proxy 198.51.100.10 port 3128`	Provides the proxy server information to the Call Home service.
Step 6	**profile CiscoTAC-1** **Example:** `Router(config-call-home)# profile CiscoTAC-1`	Enables the CiscoTAC-1 profile to be used with the Call Home service. By default, the CiscoTAC-1 profile is disabled.
Step 7	**no destination transport-method email** **Example:** `Router(config-call-home-profile)# no destination transport-method email`	Disables the email option for the Call Home service.

	Command or Action	Purpose

Step 8	**exit**
	Example:
	`Router(config-call-home-profile)# exit`
	`Router(config-call-home)#`
		Exits the Call Home destination profile configuration mode and returns to the Call Home configuration mode.

Step 9	**profile** *profile-name* **Example:** `Router(config-call-home)# profile test1`	Enters the Call Home destination profile configuration mode for the specified destination profile name. If the specified destination profile does not exist, it is created.
Step 10	**reporting smart-licensing-data** **Example:** `Router(config-call-home-profile)# reporting smart-licensing-data`	Enables data sharing with the Call Home service through the configured transport method, in this case, HTTP.
Step 11	**destination transport-method http** **Example:** `Router(config-call-home-profile)# destination transport-method http`	Enables the HTTP message transport method.
Step 12	**destination address http** *url* **Example:** `Router(config-call-home-profile)# destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService`	Connects the router to the Cisco Smart Software Manager.
Step 13	**active** **Example:** `Router(config-call-home-profile)# active`	Enables the destination profile.
Step 14	**exit** **Example:** `Router(config-call-home-profile)# exit`	Exits the Call Home destination profile configuration mode and returns to the Call Home configuration mode.

Step 15	**exit**
	Example:
	`Router(config-call-home)# exit`
	`Router(config)#`
		Exits the Call Home configuration mode and returns to the global configuration mode.

Step 16	**commit** **Example:** `Router(config)# commit`	Commits the configuration.

Connecting to CSSM On-Premise

This section describes how to configure the Call Home service for on-premise smart software through connected or disconnected mode.

SUMMARY STEPS

1. **configure**
2. **call-home**
3. **profile** *profile-name*
4. **reporting smart-licensing-data**
5. **destination transport-method http**
6. **destination address http** *url*
7. **no destination address http** *url*
8. **destination preferred-msg-format** {**long-text** | **short-text** | **xml**}
9. **active**
10. **exit**
11. **exit**
12. **http client source-interface** *ip-version interface-type interface-number*
13. **crypto ca trustpoint** *name*
14. **commit**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `Router# configure`	Enters global configuration mode.
Step 2	**call-home** **Example:** `Router(config)# call-home`	Enters Call Home configuration mode.
Step 3	**profile** *profile-name* **Example:** `Router(config-call-home)# profile test1`	Enters the Call Home destination profile configuration mode for the specified destination profile name. If the specified destination profile does not exist, it is created.
Step 4	**reporting smart-licensing-data** **Example:** `Router(config-call-home-profile)# reporting smart-licensing-data`	Enables data sharing with the Call Home service through the configured transport method, in this case, HTTP.
Step 5	**destination transport-method http** **Example:** `Router(config-call-home-profile)# destination transport-method http`	Enables the HTTP message transport method.
Step 6	**destination address http** *url* **Example:**	Configures the destination URL (CSSM) to which Call Home messages are sent.

	Command or Action	Purpose

	`Router(config-call-home-profile)# destination address http http://209.165.201.15/Transportgateway/services/DeviceRequestHandler`
	
	Or
	`Router(config-call-home-profile)# destination address http https://209.165.201.15/Transportgateway/services/DeviceRequestHandler`
		Note Ensure the IP address or the fully qualified domain name (FQDN) in the destination URL matches the IP address or the FQDN as configured for the **Host Name** on the CSSM On-Prem.

Step 7	**no destination address http** *url* **Example:** `Router(config-call-home-profile)# no destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService`	Removes the default destination address.
Step 8	**destination preferred-msg-format** {**long-text** | **short-text** | **xml**} **Example:** `Router(config-call-home-profile)# destination preferred-msg-format xml`	(Optional) Configures a preferred message format. The default message format is XML.
Step 9	**active** **Example:** `Router(config-call-home-profile)# active`	Enables the destination profile.
Step 10	**exit** **Example:** `Router(config-call-home-profile)# exit`	Exits the Call Home destination profile configuration mode and returns to the Call Home configuration mode.

Step 11	**exit**
	
	Example:
	
	`Router(config-call-home)# exit`
	`Router(config)#`
		Exits the Call Home configuration mode and returns to the global configuration mode.

Step 12	**http client source-interface** *ip-version interface-type interface-number* **Example:** `Router(config)# http client source-interface ipv4 Vlan100`	Configures a source interface for the HTTP client. **Note** This command is mandatory for a VRF interface.

Step 13	**crypto ca trustpoint** *name*
	
	Example:
	
	`Router(config)# crypto ca trustpoint SLA-TrustPoint`
	`Router(config-trustp)#`
		(Optional) Declares the trustpoint and its name.

	Command or Action	Purpose
Step 14	**commit** **Example:** `Router(config-trustp)# commit`	Commits the configuration.

Step 15	**end**
	Example:
	`Router(config-trustp)# end`
	`Router(config)#`
		Returns to the global configuration mode.

Installing CSSM On-Premise

For information on installation instructions, see the Smart Software Manager On-Prem Installation Guide.

Registering and Activating Your Router

Product registration securely associates a device with the Smart Account and the Virtual Account of your choice. It also establishes trust between the end product and the CSSM. Tokens are used to register a product with the appropriate Virtual Account on CSSM Cloud (on Cisco.com) or CSSM On-Premise.

A Registration Token:

- Can be either used once or reused multiple times. You can set a limit to the number of times a token can be reused when you create the token.
- Can be created and revoked at any time.
- Expires after a period of time (default is 30 days; minimum is one day; maximum is 365 days)

A Registration Token is not:

- Product specific: The same Registration Token can be used on different product types.
- A license, key, or PAK.
- Stored on the Cisco device and they are not persistent.
- Required after the product is registered. Token expiration has no effect on previously registered products; it simply means that that token can no longer be used to register a new product.

Generating a New Token from CSSM

Step 1 If you choose the direct cloud access deployment option, log in to CSSM from https://software.cisco.com/#.

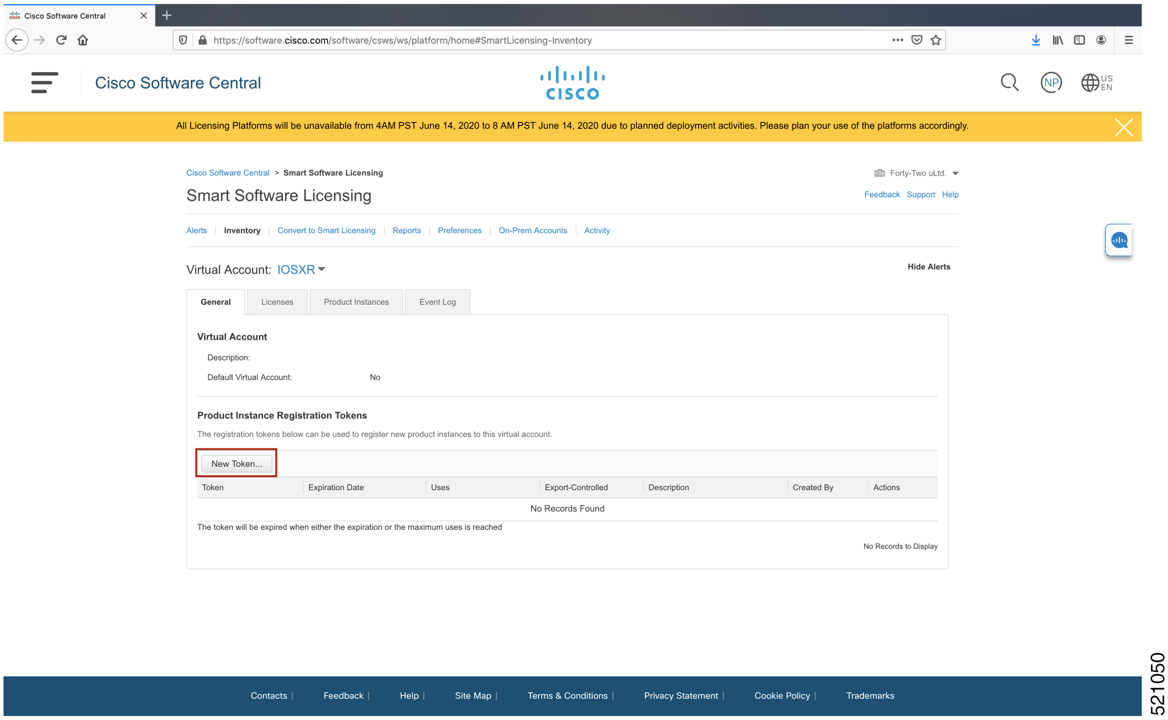
If you chose the mediated access deployment option, log in to CSSM On-Prem from *https://<on-prem-ip-address>:8443*.

Step 2 Select the **Inventory** tab.

Step 3 From the Virtual Account drop-down list, choose the virtual account to which you want to register your product.

Step 4 Select the **General** tab.

Step 5 Click **New Token**.



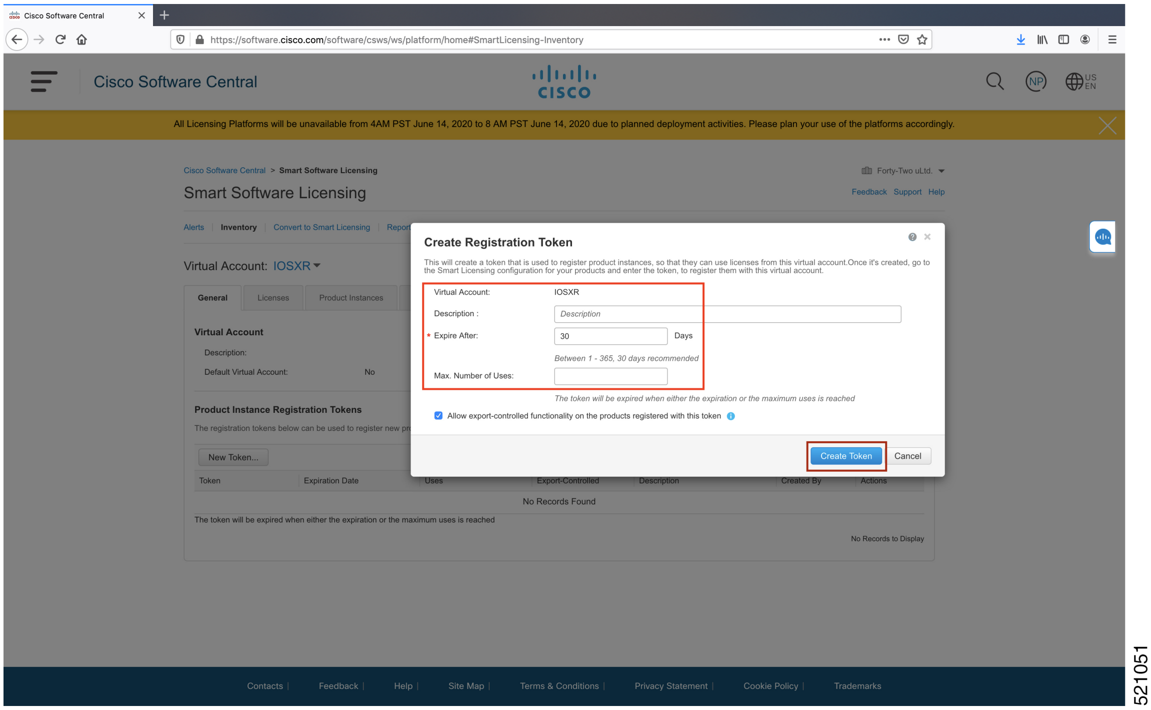
The **Create Registration Token** window is displayed.

Step 6 In the **Description** field, enter the token description.

In the **Expire After** field, enter the number of days the token must be active. The default value is 30 days.

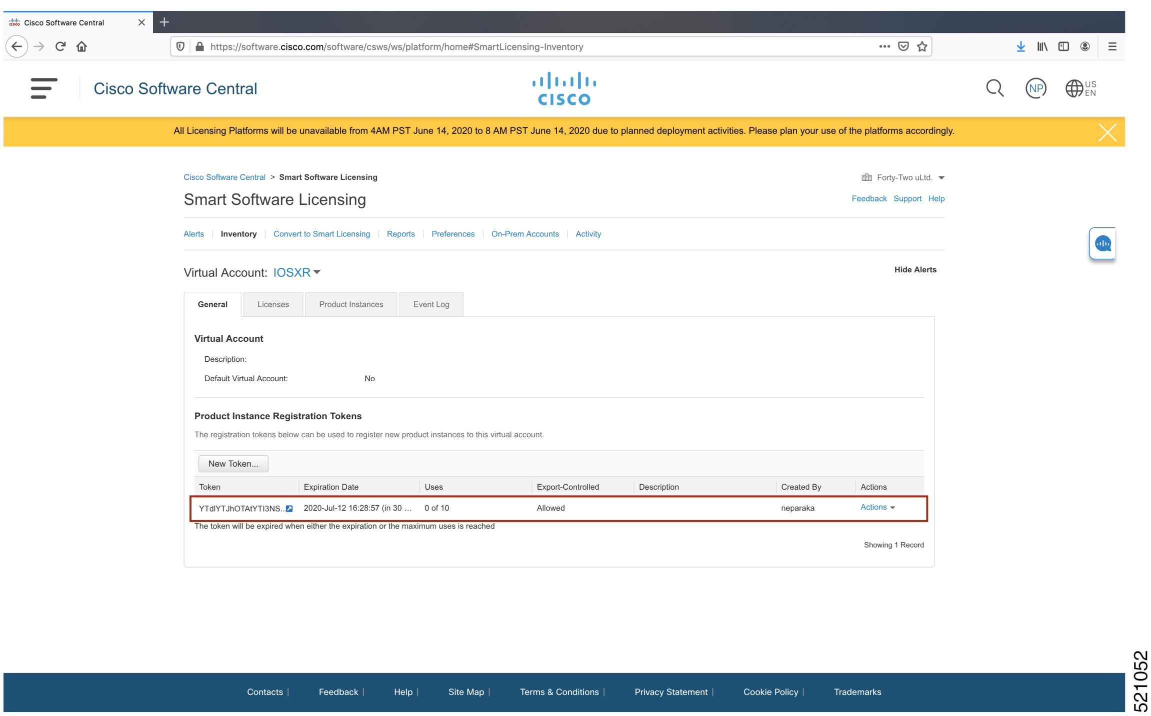
In the **Max. Number of Uses** field, enter the maximum number of uses allowed after which the token expires.

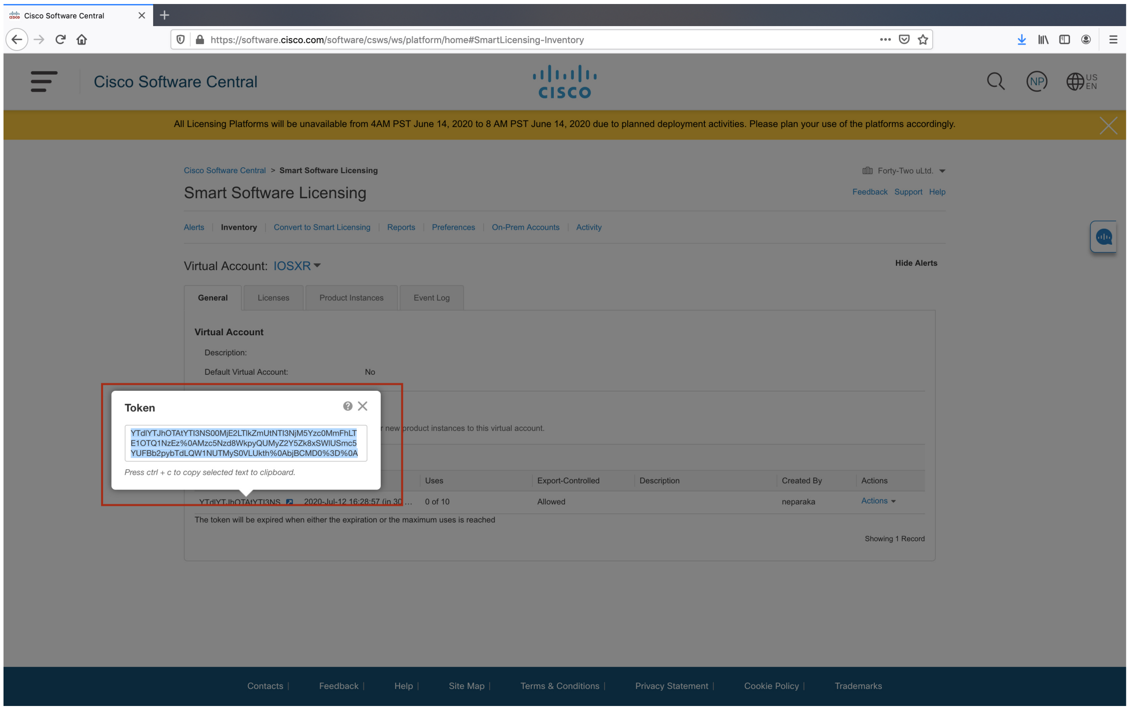
Select the **Allow export-controlled functionality on the products registered with this token** checkbox to ensure Cisco compliance with US and country-specific export policies and guidelines. For more information, see https://www.cisco.com/c/en/us/about/legal/global-export-trade.html.



Click **Create Token**.

Step 7 After the token is created, select and copy the token to a text file.





521053

You need this token to register your router.

What to do next

See the *Registering Your Device With the Token* section.

Registering Your Device With the Token

SUMMARY STEPS

1. **license smart register idtoken** *token-ID*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**license smart register idtoken** *token-ID* **Example:** `license smart register idtoken $T14UytrNXBzbEs1ck8veUtWaG5abnZJOFdDa1FwbVRa%0Ab1RMbz0%3D%0A`	Registers Smart Licensing on the router using the registration token created in the CSSM. On successful registration, the product instance is created in the CSSM virtual account and its license usage is displayed on the CSSM.

Renewing Your Smart Licensing Registration

Your registration is automatically renewed every six months. To find the status of the license, use the **license smart renew auth** command.

As long as the license is in an 'Authorized' or 'Out-of-compliance' (OOC) state, the authorization period is renewed. Grace period starts when an authorization period expires. During the grace period or when the grace period is in the 'Expired' state, the system continues to try to renew the authorization period. If a retry is successful, a new authorization period starts.

Note If the smart license renewal fails, then the product instance goes to an unidentified state and starts consuming the evaluation period.

Before you begin

Ensure that the following conditions are met to renew your smart license:

- Smart licensing is enabled.
- The router is registered.

SUMMARY STEPS

1. **license smart renew** {**auth** | **id**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**license smart renew** {**auth** | **id**} **Example:** `Router# license smart renew auth`	Renews your token ID or authorization with Cisco smart licensing.

Deregistering Your Router from CSSM

When a router is taken off the inventory, shipped elsewhere for redeployment, or returned to Cisco for replacement, you can deregister that router.

Before you begin

Ensure that a Layer 3 connection to CSSM is available to successfully deregister the device.

SUMMARY STEPS

1. **license smart deregister**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**license smart deregister** **Example:** `Router# license smart deregister`	Cancels the registration of the router and sends the router into evaluation mode. All smart licensing entitlements and certificates on the corresponding platform are removed. The product instance of the router stored on CSSM is also removed.

Enable Smart Licensing

Smart Licensing components are packaged into the asr9k mini image. The https client required for configuring the Smart Call Home is packaged into the asr9k-k9sec PIE. By default, traditional licensing mode is on. Use the steps described here to enable Smart Licensing.

> **Note** Smart Licensing on Cisco ASR 9001-S Router is not supported. Hence you should use the Product Authorization Key (PAK) to activate a license. PAK is provided when you order and purchase the right to use a feature set for a particular device. The PAK is an 11-character alphanumeric key printed on the purchase order document that is shipped with your device hardware. The PAK serves as a receipt and is an important component used in the process of obtaining, upgrading, and activating a license.
>
> For information on how to activate a license using PAK, refer Cisco ASR 9001-S 120G Upgrade License Configuration Guide.

On successful registration, the device will receive an identity certificate. This certificate is saved on your device and automatically used for all future communications with Cisco. Every 30 days, Smart Licensing will automatically renew the registration information with Cisco. If registration fails, an error will be logged. Additionally, license usage data is collected and a report is sent to you every month. If required, you can configure your Smart Call Home settings such that sensitive information (like hostname, username and password) are filtered out from the usage report.

> **Note** Once Smart Licensing mode is enabled, all CLIs related to the traditional licensing mode are disabled.

Before you begin

You must have purchased the product for which you are adding the license. When you purchase the product, you are provided with a user name and password to the Cisco Smart Software Manager portal, from where you can generate the product instance registration tokens.

SUMMARY STEPS

1. Login to Cisco Smart Software Manager at https://tools.cisco.com/rhodui/index.
2. **admin**
3. **configure**
4. **license smart enable**
5. Use the **commit** or **end** command.
6. **admin**
7. **license smart register idtoken** *token_ID*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Login to Cisco Smart Software Manager at https://tools.cisco.com/rhodui/index.	Get a token from the Cisco portal using the link. You must log in to the portal using a Cisco provided username and password. Once you have generated the token, select **Copy** hyperlink to copy the token or download the token to a text

	Command or Action	Purpose
		file. The token is used to register and activate a device, and assign the device to a virtual account. **Note** This token is valid for 30 days.
Step 2	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 3	**configure** **Example:** `RP/0/RSP0/CPU0:router(admin)# configure`	Enters administration configuration mode.

Step 4	**license smart enable**
	
	Example:
	
	`RP/0/RSP0/CPU0:router(admin-config)#license smart enable`
	`RP/0/RSP0/CPU0:router(admin-config)#show config`
	`Building configuration...`
	`!! IOS XR Configuration 5.2.0.19I`
	`license smart enable`
	`end`
		Enables basic Smart Licensing. Use the **no** form of this command to disable Smart Licensing and revert to the traditional or strict mode of licensing.

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 mode, without committing the configuration changes.
Step 6	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

Step 7	**license smart register idtoken** *token_ID*
	
	Example:
	
	`RP/0/RSP0/CPU0:router(admin)#license smart register idtoken`
	`NmE1Yzg0OWMtYmJ4`
	
	`license smart register: Registration process is`
		Use the token ID procured in step 1 to register your device.

	Command or Action	Purpose
	`in progress.Please check the syslog for the registration status and result`	

What to do next

You can use the Cisco Smart Software Manager to:

- Create virtual accounts
- Assign a registered device to a virtual account
- View licenses in a virtual account
- Manage product instance registration tokens
- Transfer a license
- View, transfer or remove product instances in a virtual account

Verify Smart Licensing Configuration

After enabling Smart Licensing, you can use the **show** commands to verify the default Smart Licensing configuration. If any issue is detected, take corrective action before making further configurations.

SUMMARY STEPS

1. **admin**
2. **show license status**
3. **show license register-status**
4. **show license entitlement**
5. **show license pool**
6. **show license cert**
7. **show license features**
8. **show license ha**
9. **show license all**
10. **exit**
11. **show call-home smart-licensing statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**show license status** **Example:** `RP/0/RSP0/CPU0:router(admin)#show license status`	Displays the compliance status of Smart Licensing. Following are the possible status: • **Waiting**: Indicates the initial state after your device has made a license entitlement request. The device establishes communication with Cisco and

	Command or Action	Purpose
		successfully registers itself with the Cisco license manager. • **Authorized**: Indicates that your device is able to communicate with the Cisco license manager, and is authorised to initiate requests for license entitlements. • **Out-Of-Compliance**: Indicates that one or more of your licenses are out-of-compliance. You must buy additional licenses. • **Eval Period**: Indicates that Smart Licencing is consuming the evaluation period. You must register the device with the Cisco Licensing manager, else your license expires. • **Grace Period**: Indicates that connectivity to the Cisco license manager is lost. You must try restore connectivity to renew the authorization period. • **Disabled**: Indicates that Smart Licensing is disabled. • **Invalid**: Indicates that Cisco does not recognize the entitlement tag as it is not in the database.
Step 3	**show license register-status** **Example:** `RP/0/RSP0/CPU0:router(admin)#show license register-status`	Displays the Smart Licensing registration status. If your registration is pending or failed, check for connectivity issues with the Cisco license manager or register the device with a new token ID.
Step 4	**show license entitlement** **Example:** `RP/0/RSP0/CPU0:router(admin)#show license entitlement`	Displays the details of the various entitlements you own.
Step 5	**show license pool** **Example:** `RP/0/RSP0/CPU0:router(admin)#show license pool`	Displays the pool to which the device belongs.
Step 6	**show license cert** **Example:** `RP/0/RSP0/CPU0:router(admin)#show license cert`	Displays details of the licensing certificate.
Step 7	**show license features** **Example:** `RP/0/RSP0/CPU0:router(admin)#show license features`	Displays the licenses that are supported on a given chassis. You can go ahead and buy the required licenses.
Step 8	**show license ha** **Example:** `RP/0/RSP0/CPU0:router(admin)#show license ha`	Displays the Smart Licensing high availability status, whether it is in active or standby mode.

	Command or Action	Purpose
Step 9	**show license all** **Example:** `RP/0/RSP0/CPU0:router(admin)#show license all`	Displays all entitlements in use. It can also be used to check if Smart Licensing is enabled. Additionally, it shows associated licensing certificates, compliance status, UDI, and other details.
Step 10	**exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits administration EXEC mode and returns to EXEC mode.
Step 11	**show call-home smart-licensing statistics**	Displays the statistics of communication between the Smart Licensing manager and the Cisco back-end using Smart Call Home. In case communication fails or drops, check your call home configuration for any errors.

The following example shows sample output from the **show call-home smart-licensing statistics**command:

```
RP/0/RSP0/CPU0:router#show call-home smart-licensing statistics
Success: Successfully sent and response received.
Failed : Failed to send or response indicated error occurred.
Inqueue: In queue waiting to be sent.
Dropped: Dropped due to incorrect call-home configuration.

Msg Subtype      Success Failed  Inqueue Dropped Last-sent (GMT-07:00)
----------------------------------------------------------------------
ENTITLEMENT      2       0       0       0       2014-04-24 18:24:34
REGISTRATION     1       0       0       0       2014-04-25 03:53:57
ACKNOWLEDGEMENT  1       0       0       0       2014-04-23 19:21:21
RENEW            1       0       0       0       2014-04-23 19:21:11
DEREGISTRATION   1       0       0       0       2014-04-25 03:31:35
```

Renew Smart Licensing Registration

In general, your registration is automatically renewed every 30 days. Use this option to make an on-demand manual update of your registration. Thus, instead of waiting 30 days for the next registration renewal cycle, you can issue this command to instantly find out the status of your license.

Before you begin

You must ensure that the following conditions are met to renew your smart license:

- Smart licensing is enabled.
- The device is registered.

SUMMARY STEPS

1. **admin**
2. **license smart renew** {**auth** | **id**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

Step 2	**license smart renew** {**auth** | **id**}
	Example:
	`RP/0/RSP0/CPU0:ROA(admin)#license smart renew auth`
	`Tue  Apr 22 09:12:37.086 PST`
	`license smart renew auth: Authorization process is in progress.`
	`Please check the syslog for the authorization status and result.`
		Renew your ID or authorization with Cisco smart licensing. If ID certification renewal fails, then the product instance goes to an unidentified state and starts consuming the evaluation period.
		Note Authorization periods are renewed by the Smart Licensing system every 30 days. As long as the license is in an 'Authorized' or 'Out-of-compliance' (OOC), the authorization period is renewed. Grace period starts when an authorization period expires. During the grace period or when the grace period is in the 'Expired' state, the system continues to try renew the authorization period. If a retry is successful, a new authorization period starts.

De-register Smart Licensing

When your device is taken off the inventory, shipped elsewhere for redeployment or returned to Cisco for replacement using the return merchandise authorization (RMA) process, you can use the de-register option to cancel the registration on your device. Use the following steps to cancel device registration:

SUMMARY STEPS

1. **admin**
2. **license smart deregister**

DETAILED STEPS

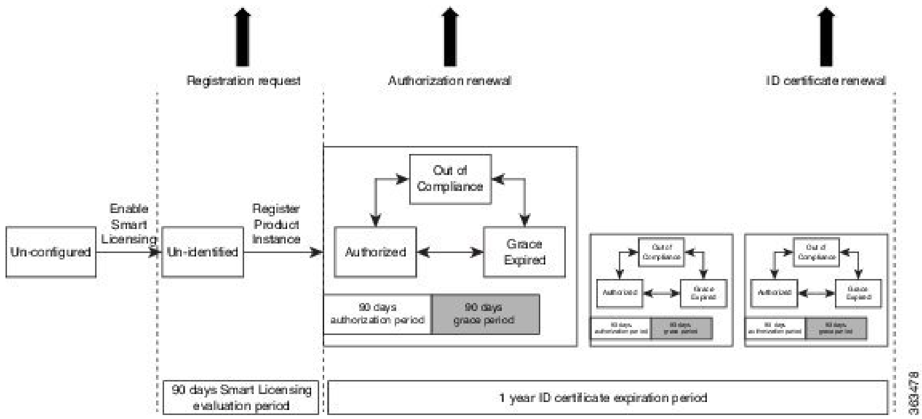
	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

Step 2	**license smart deregister**
	Example:
	`RP/0/RSP0/CPU0:IMC0(admin)#license smart deregister`
	`license smart deregister: Success`
		Cancels the device registration, and sends it into a 30-day evaluation mode. All Smart Licensing entitlements and certificates on the platform are removed.
		Note Though the product instance has been de-registered from the Cisco license cloud service, Smart Licencing is still enabled.

	Command or Action	Purpose
	`License command "license smart deregister " completed successfully.`	

Smart Licensing Workflow

The Smart Licensing workflow is depicted in this flowchart.



Cisco Smart Software Manager Overview

Cisco Smart Software Manager enables you to manage all of your Cisco Smart software licenses from one centralized website. With Cisco Smart Software Manager, you organize and view your licenses in groups called virtual accounts (collections of licenses and product instances). Use the Cisco Smart Software Manager to do the following tasks:

- Create, manage or view virtual accounts.
- Create and manage Product Instance Registration Tokens.
- Transfer licenses between virtual accounts or view licenses.
- Transfer, remove or view product instances.
- Run reports against your virtual accounts.
- Modify your email notification settings.
- View overall account information.

The Cisco Smart Software Manager **Help** describes the procedures for carrying out these tasks. You can access the Cisco Smart Software Manager on https://webapps.cisco.com/software/csws/ws/platform/home, by clicking **Licensing**, and then selecting **Smart Software Manager**; and then login using the username and password provided by Cisco.

Note Use Chrome 32.0, Firefox 25.0 or Safari 6.0.5 web browsers to access the Cisco Smart Software Manager. Also, ensure that Javascript 1.5 or a later version is enabled in your browser.

Licenses, Product Instances, and Registration Tokens

Licenses

Licenses are required for all Cisco products. All Cisco product licenses are one of two types which vary depending on the product:

- Perpetual licenses—Licenses that do not expire.
- Term licenses—Licenses that automatically expire after a set amount of time: one year, three years, or whatever term was purchased.

In addition, there are demo licenses that expire after at most 60 days. As implied by the name, demo licenses are not intended for production use.

All product licenses reside in a virtual account.

Product Instances

A product instance is an individual device with a unique device identifier (UDI) that is registered using a product instance registration token (or registration token). You can register any number of instances of a product with a single registration token. Each product instance can have one or more licenses residing in the same virtual account. Product instances must periodically connect to the Cisco Smart Software Manager servers during a specific renewal period. If a product instance fails to connect, it is marked as having a license shortage, but continues to use the license. If you remove the product instance, its licenses are released and made available within the virtual account.

Product Instance Registration Tokens

A product requires a registration token until you have registered the product. Registration tokens are stored in the Product Instance Registration Token Table associated with your enterprise account. Once the product is registered the registration token is no longer necessary and can be revoked and removed from the table without effect. Registration tokens can be valid from 1 to 365 days.

Virtual Accounts

Smart Licencing allows you to create multiple license pools or virtual accounts within the Smart Software Manager portal. Using the **Virtual Accounts** option you can aggregate licenses into discrete bundles associated with a cost center so that one section of an organization cannot use the licenses of another section of the organization. For example, if you segregate your company into different geographic regions, you can create a virtual account for each region to hold the licenses and product instances for that region.

All new licenses and product instances are placed in the default virtual account in the Smart Software Manager, unless you specify a different one during the order process. Once in the default account, you may choose to transfer them to any other account as desired, provided you have the required access permissions. See Licenses, Product Instances, and Registration Tokens, on page 355 for details.

Use the Smart Software Manager portal at https://tools.cisco.com/rhodui/index to create license pools or transfer licenses.

Compliance reporting

On a periodic basis, as described by the terms of the Smart Licensing contract, reports are automatically sent to you containing inventory and license compliance data. These reports will take one of three forms:

- **Periodic Record:** This record is generated on a periodic (configurable) basis with relevant inventory data saved at a given point of time. This report is saved within the Cisco cloud for archival.
- **Manual Record:** You can manually generate this record with relevant inventory data saved at any given point of time. This report will be saved within the Cisco cloud for archival.

- **Compliance Warning Report:** This report is automatically or manually generated when a license compliance event occurs. This report does not contain a full inventory data, but only any shortfalls in entitlements for a given software license.

  You can view these reports from the Smart Software Manager portal at https://tools.cisco.com/rhodui/index.

Traditional Licensing Consideration in Smart Licensing

Traditional licensing, and the associated commands, currently co-exist with Smart Licensing. By default, the software image is loaded with the traditional, strictly-enforced mode of licensing. You may want to retain the traditional licensing model in the following scenarios:

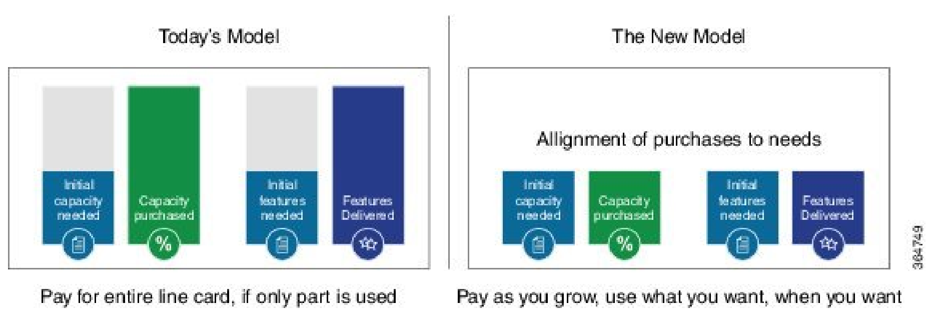
- when there are multiple users, and you do not know the actual end user of your software.
- when the software is deployed in a location with limited access to the license and inventory management solution.
- when the user has opted not to establish a Smart Call Home relationship with Cisco.
- when a Smart Call Home relationship cannot be maintained with the user owing to logistics and a fallback is required.

Note All traditional licencing CLI commands are disabled if Smart Licensing is enabled. However, you can continue to access the traditional licenses stored under: /disk0:/license/*. Certificates used by Smart Licencing are located under /disk0:/sla/. Respective CLIs are restored when licensing schemes are switched.

Consumption Model

The consumption model is a new pricing model for line cards to align the initial purchase to your actual needs. This model provides the ability to deploy a line card on day 1 with minimum ports activated to meet the current traffic demands. Over time as the traffic grows, you can add additional ports in 10G port increments. This provides a flexible deployment model with the ability to increase bandwidth to meet your demands.

Figure 11: Comparison - Current Purchasing Model And The New Consumption Model



> **Note**
>
> The consumption model line cards require the users to deploy Smart Licensing to help track and provide visibility into license usage across their network. For information on Smart Licensing see Information About Smart Licensing, on page 329.
>
> You must have a Smart Account created to place an order for the consumption model line card. You can create your Cisco Smart Account at this webpage: https://webapps.cisco.com/software/company/smartaccounts/home#accountcreation-account/request. For information on how to create a Cisco Smart Account, see: http://www.cisco.com/c/en/us/products/collateral/software/one-software/solution-overview-c22-733273.html.

Supported Consumption Model Line Cards

The line cards that can be deployed using consumption model are referred to as the consumption model line cards. The supported consumption model line cards are:

- A9K-8X100GE-CM
- A99-8X100GE-CM
- A99-12X100GE-CM
- A9K-MOD400-CM

Ordering the Consumption Model Line Card using the Consumption Model

The three steps involved in ordering a consumption model line card using the Consumption Model are:

Figure 12: Steps involved in ordering the consumption model line card using the Consumption Model

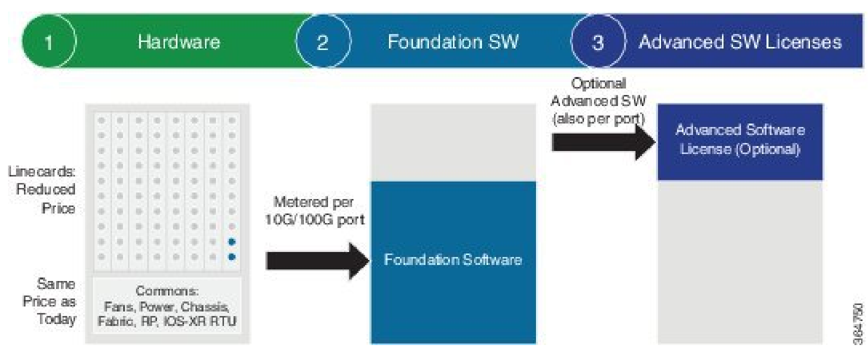
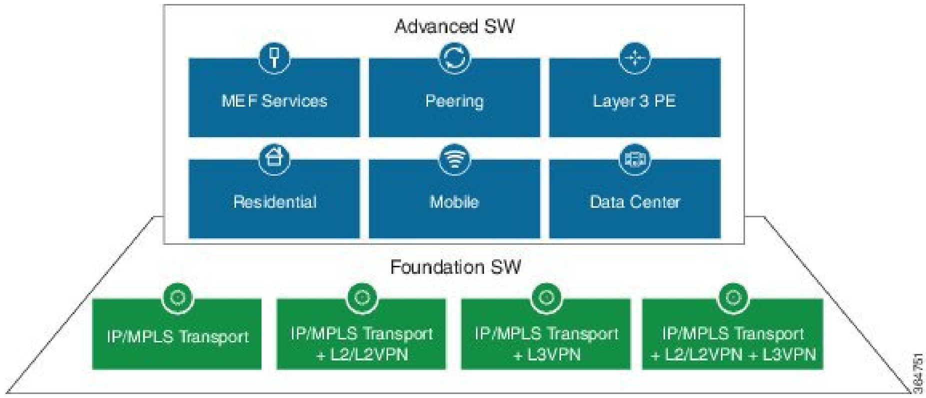


Figure 13: Foundation Software Licenses And Advanced Software Licenses



1. Choose the hardware: Select a line card that supports Consumption Model.

   The consumption model line cards require Smart Licensing to be installed at your location to function. When placing an order, you must enter their Smart Account information. For more information on how to create a Cisco Smart Account, see: http://www.cisco.com/c/en/us/products/collateral/software/one-software/solution-overview-c22-733273.html.

   Table 42: Minimum Foundation Software Licenses for the CM line cards

PID	Minimum Foundation Software License Required
A9K-8X100GE-CM	300G (30 x 10G)
A99-8X100GE-CM	300G (30 x 10G)
A99-12X100GE-CM	400G (40 x 10G)
A9K-MOD400-CM-BUN	200G (20 x 10G)

2. Choose your foundation software licenses: This provides the transport protocol (IP/MPLS, L2VPN, L3VPN, or L2VPN and L3VPN) as well as "per 10G" port activation.

   Pick your foundation software licenses based on the feature set and the scale required. The following is the list of the available licenses. The licenses listed are per 10G RTU (Right to Use) and are required to activate a port.

   Table 43: Foundation Software Licenses for Cisco IOS XR and Cisco IOS XR 64 bit

Foundation License PID	Description
S-A9K-IPB-10G	ASR 9000 IP/MPLS Basic 10G Foundational License
S-A9K-IPP-10G	ASR 9000 IP/MPLS Premium 10G Foundational License
S-A9K-L2B-10G	ASR 9000 IP/MPLS/L2VPN Basic 10G Foundational License
S-A9K-L2P-10G	ASR 9000 IP/MPLS/L2VPN Premium 10G Foundational License
S-A9K-L3B-10G	ASR 9000 IP/MPLS/L3VPN Basic 10G Foundational License
S-A9K-L3P-10G	ASR 9000 IP/MPLS/L3VPN Premium 10G Foundational License
S-A9K-L2L3B-10G	ASR 9000 IP/MPLS/L2VPN/L3VPN Basic 10G Foundational License

Foundation License PID	Description
S-A9K-L2L3P-10G	ASR 9000 IP/MPLS/L2VPN/L3VPN Premium 10G Foundational License

3. Choose your advanced software licenses: This provides the advanced feature support such as hierarchical QoS, OAM (Operations, Administration, and Maintenance), and virtual interfaces.

   Pick your advanced software licenses, optional, you can select one or more of them from the following list. The licenses listed are per 10G RTU (Right to Use).

   Below tables list supported advance software licenses for Cisco IOS XR and Cisco IOS XR 64 bit respectively:

Table 44: Advanced Software Licenses for Cisco IOS XR

Advanced Software License PID	Description
S-A9K-HQOS-RTU-10	ASR 9000 H-QoS 10G Right to Use License
S-A9K-MAP-RTU-10	ASR 9000 CGN Stateless MAP 10G Right to Use License
S-A9K-OAM-RTU-10	ASR 9000 OAM 10G Right to Use License
S-A9K-VIRT-RTU-10	ASR 9000 Virtual Interfaces 10G Right to Use License
S-A9K-EVPN-RTU-10	ASR 9000 E-VPN 10G Right to Use License
S-A9K-VXLN-RTU-10	ASR 9000 VxLAN 10G Right to Use License
S-A9K-DWDM-RTU-10	ASR 9000 IPoDWDM 10G Right to Use License
S-A9K-MAC-RTU-10	ASR 9000 MACSec 10G Right to Use License
S-A9K-MAC-RTU-40	ASR 9000 MACSec 40G (4x10G) Right to Use License
S-A9K-MAC-RTU-100	ASR 9000 MACSec 100G (10x10G) Right to Use License

Table 45: Advanced Software Licenses for Cisco IOS XR 64 bit

Advanced Software License PID	Description
S-A9K-HQOS-RTU-10	ASR 9000 H-QoS 10G Right to Use License
S-A9K-OAM-RTU-10	ASR 9000 OAM 10G Right to Use License
S-A9K-VIRT-RTU-10	ASR 9000 Virtual Interfaces 10G Right to Use License
S-A9K-EVPN-RTU-10	ASR 9000 E-VPN 10G Right to Use License
S-A9K-VXLN-RTU-10	ASR 9000 VxLAN 10G Right to Use License
S-A9K-DWDM-RTU-10	ASR 9000 IPoDWDM 10G Right to Use License
S-A9K-MAC-RTU-10	ASR 9000 MACSec 10G Right to Use License

Advanced Software License PID	Description
A9K-NVSAT1-LIC	ASR 9000 NV Satellite Right to Use License
A9K-LI-LIC	ASR 9000 Lawful Intercept Right to Use License
A9K-MOBILE-LIC	ASR 9000 Timing Advanced Mobile License Right to Use License

Table 46: Perpetual Licensing (BAU) Usage Pattern

License Name	Description	Hardware Supported	Consumption Pattern
S-A9K-24P10G-IVRF	VRF licence for upto 8 VRF instances per 24-port 10G/1G	A9K-24X10GE-1G-SE, A9K-24X10GE-1G-TR, A9K-48X10GE-1G-SE, and A9K-48X10GE-1G-TR	If number of VRFs are less than or equal to 8, IVRF license is consumed. If number of VRFs are greater than 8, AIP license is consumed. If customer has less than or equal to 8 VRFs, who purchased AIP license in their smart account, please contact your Cisco account representative if you see issues like insufficient IVRF licenses.
S-A9K-24P10G-AIP-TR	Advanced IP Licence for full scale VRFs for 24-port 10G A9K-24P10GAIPTR /1G TR LC		
S-A9K-24P10G-AIP-SE	Advanced IP Licence for full scale VRFs for 24-port 10G/1G SE LC		
S-A9K-48P10G-IVRF	Infrastructure VRF licence for upto 8 VRF instances per 48-port 10G/1G		
S-A9K-48P10G-AIP-TR	Advanced IP Licence for full scale VRFs for 48-port 10G/1G TR LC		
S-A9K-48P10G-AIP-SE	Advanced IP Licence for full scale VRFs for 48-port 10G/1G SE LC		
S-A9K-48P1G-AIP-TR	Advanced IP Licence for full scale VRFs for 48-port 1G mode TR LC		
S-A9K-48P1G-AIP-SE	Advanced IP Licence for full scale VRFs for 48-port 1G mode SE LC		
S-A9K-48P10G-TR-UG	48-port 1G to 10G upgrade license for TR LC		
S-A9K-48P10G-SE-UG	48-port 1G to 10G upgrade license for SE LC		
S-A9K-24P-80G-RTU-SE	ASR9K 80G Upgrade license for 24-port 10G/1G dual rate SE LC		
S-A9K-24P-80G-RTU-TR	ASR9K 80G Upgrade license for 24-port 10G/1G dual rate TR LC		

License Name	Description	Hardware Supported	Consumption Pattern
			In 24 port PG, 80G licenses is consumed as shown below: • Capacity upto 80G – Consume 1x A9K-24P-80G-RTU-SE/ A9K-24P-80G-RTU-TR • Capacity upto 160G – Consume 2x A9K-24P-80G-RTU-SE/ A9K-24P-80G-RTU-TR • Capacity upto 240G – Consume 3x A9K-24P-80G-RTU-SE/ A9K-24P-80G-RTU-TR

Configuration Examples:

The Consumption Model line cards provide the flexibility to configure the line card on a per 10G port basis. Here are a few examples of configurations of the existing TR and SE versions of line cards using the Consumption Model.

TR Equivalent Configuration

The TR equivalent configuration is a configuration with L2 Premium Foundation Software License plus OAM and Virtual Advanced Software licenses.

Line Card PID	Description	Quantity
A9K-8X100GE-CM A99-8X100GE-CM A9K-20X10GE-CM A99-12X100GE-CM	ASR 9000 8-port 100GE Consumption Model Line Card ASR 9900 8-port 100GE Consumption Model Line Card ASR 9000 20-port 10GE Consumption Model Line Card ASR 9900 12-port 100GE Consumption Model Line Card	1
Foundation Software PID	**Scale**	**Quantity of 10G Licenses**
S-A9K-L2P-10G	L2-P (Layer 2 Premium Foundation Software License)	30
Advanced Software PID	**Description**	**Quantity**
S-A9K-OAM-RTU-10	ASR 9000 OAM 10Gbps Right to Use License	30

Line Card PID	Description	Quantity
S-A9K-VIRT-RTU-10	ASR 9000 Virtual Interfaces 10Gbps Right to Use License	30

SE Equivalent Configuration

The SE equivalent configuration is a configuration with L2 Premium Foundation Software License plus H-QoS, OAM, and Virtual Advanced Software licenses.

Line Card PID	Description	Quantity
A9K-8X100GE-CM A99-8X100GE-CM	ASR 9000 8-port 100GE Consumption Model Line Card ASR 9900 8-port 100GE Consumption Model Line Card	1
Foundation Software PID	**Scale**	**Quantity of 10G Licenses**
S-A9K-L2P-10G	L2-P (Layer 2 Premium Foundation Software License)	30
Advanced Software PID	**Description**	**Quantity**
S-A9K-HQOS-RTU-10	ASR 9000 H-QoS 10Gbps Right to Use License	30
S-A9K-OAM-RTU-10	ASR 9000 OAM 10Gbps Right to Use License	30
S-A9K-VIRT-RTU-10	ASR 9000 Virtual Interfaces 10Gbps Right to Use License	30

Implementing Default Licensing

Prerequisites for Configuring Software Entitlement

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 Default (Traditional) Licensing

To configure software license entitlements using the default mode of licensing, you need to understand the concepts described in this module.

Types of Licenses

The following types of licenses are currently defined:

- Permanent licenses—Licenses that enable a designated feature permanently, as long as the license resides on the router.

Router License Pools

License pools are maintained by the router. By default, all added licenses are allocated to the owner secure domain router (SDR) license pool, and they can be freely allocated to any slot in the router. Features on cards belonging to the owner SDR are granted licenses based on availability in the owner SDR license pool.

Chassis-Locked Licenses

Licenses are locked to a unique device identifier (UDI). The UDI is comprised of the chassis serial number, along with an additional identifier. The complete set of UDI information can be displayed using the **show license udi** command. The license manager parses the user-provided license and verifies that it is valid for the chassis it is running on and determines if the license is being readded.

Slot-Based Licenses

Feature licenses are allocated to router slots and not cards. Therefore, if a card is replaced, the existing license is applied to the newly inserted card. For example, if you have eight licenses for Layer 3 VPN in the system, you can configure Layer 3 VPN features on any eight cards in the router, and the licenses are allocated to the slots within which the cards are installed. If a card is removed from one of these licensed slots, say slot 3, and entered into an empty slot with no license, say slot 5, the license remains with slot 3 and the feature cannot be activated on slot 5 with the permanent license entered earlier by the user. In this case, you can release the license to the appropriate license pool by removing the configuration of the card (while it is inserted), or by using the **license move slot** command. When you configure the feature on slot 5, the license is checked out.

Configure Licenses Using Default Licensing

Adding a License for a New Feature

This task describes how to acquire a permanent license for a feature that you have purchased or an evaluation license for a feature that you have arranged with your sales representative to try. Use this procedure to replace evaluation licenses with permanent licenses.

Before you begin

You must have purchased the feature for which you are adding the license. When you purchase the feature, you are provided with a product authorization key (PAK) that you use to download the license.

SUMMARY STEPS

1. **admin**
2. **show license udi**
3. http://www.cisco.com/go/license
4. Copy the license to your TFTP server.

5. **admin**
6. **license add** *license-name* [**sdr** *sdr-name*]
7. **configure**
8. **license** *license-name* **location** {**all** | *node-id*}
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

Step 2	**show license udi**
	Example:
	`RP/0/RSP0/CPU0:router(admin)# show license udi`
	`Mon Jul 13 04:36:32.715 PST`
	`Local Chassis UDI Information:`
	`  PID         : ASR-9010-AC`
	`  S/N         : FOX1232H67M`
	`  Operation ID: 1`
		Displays the UDI of the chassis. This consists of a product identifier (PID), serial number (S/N), and operation identifier (Operation ID).

Step 3	http://www.cisco.com/go/license	Go to the license tool on Cisco.com. You must log in to the site before you can access the license tool. Follow the instructions for product license registration. You are required to enter the feature PAK and the chassis UDI to acquire the license. **Note** If you are installing a permanent license, you should have received the PAK when you purchased the feature. If you are installing an evaluation license, your sales representative should provide you with the PAK.
Step 4	Copy the license to your TFTP server.	You will be issued a license. You can copy the license and store it on your computer, or alternatively, you can request that the license be sent to you in an e-mail. When you have received the license, copy it to a TFTP server that is accessible by your router.
Step 5	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.

	Command or Action	Purpose
Step 6	**license add** *license-name* [**sdr** *sdr-name*] **Example:** `RP/0/RSP0/CPU0:router(admin)# license add tftp://192.10.10.10/mylicenses/lc40g_lic`	Adds the license to the SDR license pool. By default, the license is added to the owner SDR license pool.
Step 7	**configure** **Example:** `RP/0/RSP0/CPU0:router(admin)# configure`	Enters administration configuration mode.
Step 8	**license** *license-name* **location** {**all** | *node-id*} **Example:** `RP/0/RSP0/CPU0:router(admin-config)# `**`license A9K-ADV-OPTIC-LIC location 0/0/CPU0`**	(Optional) Binds the license to the slot where it is to be used. **Note** Beginning with Cisco IOS XR Release 4.3.1, this command is optional. If you do not use this command, it is configured as though the license is bound to all slots.
Step 9	**exit** **Example:** `RP/0/RSP0/CPU0:router(admin)# exit`	Exits administration EXEC mode.

What to do next

To use the feature associated with the added license, you must configure it on your router. To configure Layer 3 VPN, see the *Implementing MPLS Layer 3 VPNs on Cisco IOS XR Software* module in *MPLS Configuration Guide for Cisco ASR 9000 Series Routers*.

To verify that your Layer 3 VPN configuration is operational, use the **show rsi interface all global** command.

Backing Up Licenses

When your router is configured with the licenses that you require, you should perform this task to back up all licenses. Backing up licenses makes it easier to restore them if there is a problem.

SUMMARY STEPS

1. **admin**
2. **license backup** *backup-file*
3. **show license backup** *backup-file*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:**	Enters administration EXEC mode.

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

Step 2	**license backup** *backup-file*
	Example:
	`RP/0/RSP0/CPU0:router(admin)# license backup disk1:/license_back`
	`License command "license backup disk1:/license_back" completed successfully.`
		Backs up all licenses on the router to a backup file in the specified location. The backup file can be a local file or a remote file on a TFTP or RCP server.

Step 3	**show license backup** *backup-file* **Example:** `RP/0/RSP0/CPU0:router(admin)# show license backup disk1:/license_back`	Displays the contents of the backup file.

Examples

The following example shows sample output from the **show license backup** command.

```
RP/0/RSP0/CPU0:router(admin)# show license backup disk1:/license_back

Local Chassis UDI Information:
  S/N         : TBA09370035
  Operation ID:   5
Licenses      :
FeatureID    Type                     #installed
CRS-MSC-40G  Slot based,    Permanent          2
XC-L3VPN     Slot based,    Permanent          1


RP/0/RSP0/CPU0:router(admin)# show license backup disk0:/lic_backup.pkg

Tue Jul 27 17:12:44.982 pst

Local Chassis UDI Information:
  S/N         : FOX1316G5TL
  Operation ID:   9

  FeatureID: A9K-ADV-OPTIC-LIC (Slot based, Permanent)
  Total licenses  1
    Pool: Owner  1
    Allocated Node(s):
      0/0/CPU0 1 [Owner]

  FeatureID: A9K-ADV-VIDEO-LIC (Slot based, Evaluation)
  Total licenses  1
    Pool: Owner  1
    Allocated Node(s):
      0/RSP0/CPU0 1 [Owner]
```

```
FeatureID: A9K-iVRF-LIC (Slot based, Permanent)
Total licenses  1
  Pool: Owner  1

FeatureID: A9K-iVRF-LIC (Slot based, Evaluation)
Total licenses  3
  Pool: Owner  3
  Allocated Node(s):
    0/1/CPU0 1 [Owner]
```

Restoring Licenses

If your licenses become corrupted, and you have previously created a backup of your licenses, you can perform this task to restore the licenses to your router.

Before you begin

You must have created a backup file of your licenses before you can restore them on your router.

SUMMARY STEPS

1. **admin**
2. **show license backup** *backup-file*
3. **license restore** *backup-file*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**admin** **Example:** `RP/0/RSP0/CPU0:router# admin`	Enters administration EXEC mode.
Step 2	**show license backup** *backup-file* **Example:** `RP/0/RSP0/CPU0:router(admin)# show license backup disk1:/license_back`	Displays the contents of the backup file. You should verify the contents of the backup file before you restore your licenses.
Step 3	**license restore** *backup-file* **Example:** `RP/0/RSP0/CPU0:router(admin)# license restore disk1:/license_back`	Restores all licenses on the router from a backup file in the specified location. This can be a local file, or a remote file on a TFTP or RCP server.

Examples

This example shows sample output from the **license restore** command.

```
RP/0/RSP0/CPU0:router(admin)# license restore disk1:/license_back

Info: This command will erase all existing licenses.
Info: It is strongly recommended to backup existing licenses first.
```

```
Do you wish to proceed? [yes/no]: y

License command "license restore disk1:/license_back" completed successfully.
```

Troubleshooting License Issues after a Software Upgrade

In the instance that you were running Cisco IOS XR Release 3.9.0 and had the optic feature enabled on a interface and the A9K-ADV-OPTIC-LIC license was active on a particular slot, when you upgrade to Cisco IOS XR Release 4.0.0, the A9K-ADV-OPTIC-LIC license is still active, but you may get the following warning message:

```
RP/0/RSP0/CPU0:Jul 27 14:22:22.594 : licmgr[236]:
%LICENSE-LICMGR-4-PACKAGE_LOCATION_LICENSE_INVALID :
Feature associated to package A9K-ADV-OPTIC-LIC configured
on node 0/4/CPU0 without a valid license
```

To solve this issue, configure the **license** command in administration EXEC mode. This binds the A9K-ADV-OPTIC-LIC license to the slot on which you are using the license. For example:

```
RP/0/RSP0/CPU0:router(admin-config)# license A9K-ADV-OPTIC-LIC location 0/4/CPU0
RP/0/RSP0/CPU0:router(admin-config)# commit
```

CHAPTER 20

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.

SyncE Hardware Support Matrix

This table provides details on the harware that supportes 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 47: 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
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

Hardware Variant	Cisco IOS XR	Cisco IOS XR 64 bit
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.

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.

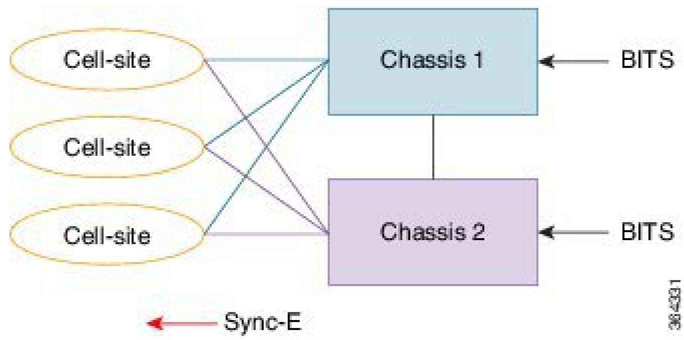
ICS-Time-of-Day

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:

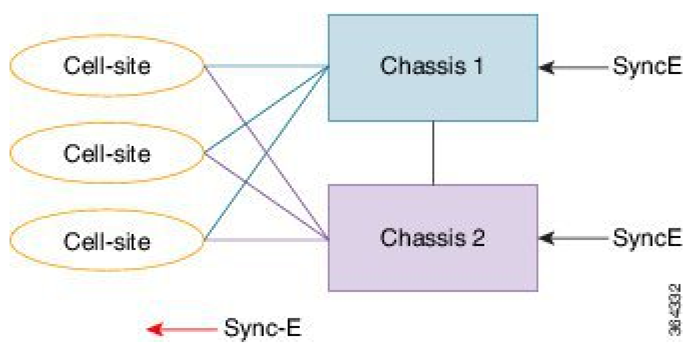
Figure 14: No inter-chassis frequency support



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 15: 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.

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.
Step 3	**clock-interface timing-mode** {**independent** | **system** **Example:** `RP/0/RSP0/CPU0:router(config-freqsync)# clock-interface timing-mode system`	(Optional) 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.

	Command or Action	Purpose
		• **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**
	Example:
	`RP/0/RSP0/CPU0:router(config-freqsync)# end`
	or
	`RP/0/RSP0/CPU0:router(config-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.

	Command or Action	Purpose
		• 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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
		• **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**
	
	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.

	Command or Action	Purpose
		• 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.

	Command or Action	Purpose
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.
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.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td></td><td>• For SONET and clock interfaces, this causes DNUs to be sent, and ignores any received QL value.</td></tr>
<tr><td>Step 10</td><td>time-of-day-priority priority
Example:
<code>RP/0/RSP0/CPU0:router(config-if-freqsync)# time-of-day-priority 50</code></td><td>(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.</td></tr>
<tr><td>Step 11</td><td>quality transmit {exact | highest | lowest} itu-t option ql-option
Example:
<code>RP/0/RSP0/CPU0:router(config-clk-freqsync)# quality transmit
highest itu-t option 1 prc</code></td><td>(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.</td></tr>
<tr><td>Step 12</td><td>quality receive {exact | highest | lowest} itu-t option ql-option
Example:
<code>RP/0/RSP0/CPU0:router(config-clk-freqsync)# quality receive
highest itu-t option 1 prc</code></td><td>(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.</td></tr>
<tr><td>Step 13</td><td>Use one of these commands:</td><td>Saves configuration changes.</td></tr>
</table>

	Command or Action	Purpose

	• **end**
	• **commit**
	Example:
	`RP/0/RSP0/CPU0:router(config-if-freqsync)# end`
	or
	`RP/0/RSP0/CPU0:router(config-if-freqsync)# commit`
		• 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.

	Command or Action	Purpose
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.
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 377
- 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.

CHAPTER 21

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 48: 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.

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
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:

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.
- Cisco ASR 9000 Series Routers do not support Class B 1 Pulse Per Second (PPS) performance with Forward Error Correction (FEC) enabled optics.
- One-step clock operation on G.8275.1 profile is not supported on a PTP server.
- 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
```

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 22

Configuring Zero Touch Provisioning

Zero Touch Provisioning (ZTP) works as a Third Party App (TPA) in Route-Switch Processor (RSP) and Route Processor (RP). ZTP was designed to perform two different operations:

- Download and apply an initial configuration.
- Download and execute a shell script.

ZTP works as following:

1. XR scripts that run on boot, invoke DHCP request.
2. DHCP server returns a user script.
3. User script then provisions router.

Prior to Cisco IOS XR Release 6.1.1, ZTP was executed within the default network namespace and could not access the data interfaces directly. Starting with Cisco IOS XR Release 6.1.1, ZTP is executed inside the global Virtual Routing and Forwarding (VRF) network namespace with full access to all the data interfaces.

Note ZTP functionality and commands are available on XR 64 Bit only for Cisco ASR9000.

ZTP requires two external services: a DHCP server and an HTTP server. ZTP is launched from Cisco IOS XR process manager when the system reaches the last process to be scheduled for execution. At the beginning of its execution, ZTP will scan the configuration for the presence of a username. If there are no username configured, ZTP will invoke a DHCP client on the management interface for IPv4 and IPv6 simultaneously, and wait for a response.

This module contains the following topics:

Manual ZTP Invocation

Manual Zero Touch Provisioning (ZTP) can be invoked manually via CLI commands. This manual way helps you to provision the router in stages. Ideal for testing out ZTP configuration without a reboot. If you would like to invoke a ZTP on an interfaces(data ports or management port), you don't have to bring up and configure the interface first. You can execute the **ztp initiate** command, even if the interface is down, ZTP script will bring it up and invoke dhclient. So ZTP could run over all interfaces no matter it is up or down.

Use the **ztp initiate**, **ztp breakout**, **ztp terminate**, **ztp enable, ztp disable**, and **ztp clean** commands to force ZTP to run over more interfaces.

- **ztp initiate**— Invokes a new ZTP DHCP session. Logs can be found in **/disk0:/ztp/ztp.log**.
- **ztp terminate**—Terminates any ZTP session in progress.
- **ztp enable**—Enables ZTP at boot.
- **ztp disable**—Disables ZTP at boot.
- **ztp clean**—Removes only the ZTP state files.

  From release 6.2.3, the log file ztp.log is saved in **/var/log** folder, and a copy of log file is available at **/disk0:/ztp/ztp.log** location using a soft link. However, executing **ztp clean** clears files saved on disk and not on **/var/log** folder where current ZTP logs are saved. In order to have a log from current ZTP run, you must manually clear the ZTP log file from **/var/log/** folder.

For more information of the commands, see the ZTP command chapter in the *System Management Command Reference for Cisco ASR 9000 Series Routers*.

This task shows the most common use case of manual ZTP invocation: invoke 10x10 breakout discovery and ZTP.

SUMMARY STEPS

1. **ztp breakout**
2. **ztp initiate dataport**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ztp breakout** **Example:** `RP/0/RSP0/CPU0:router# ztp breakout`	ZTP will enable breakout ports.
Step 2	**ztp initiate dataport** **Example:** `RP/0/RSP0/CPU0:router# ztp initiate dataport`	Invoke DHCP sessions on all dataport or Line Card interfaces found. ZTP runs in the background. Please use **show logging** or look at **/disk0:/ztp/ztp.log** to check progress.

Authentication on Data Ports

On fresh boot, ZTP process is initiated from management ports and may switch to data ports. To validate the connection with DHCP server, authentication is performed on data ports through DHCP option 43 for IPv4 and option 17 for IPv6. These DHCP options are defined in option space and are included within **dhcpd.conf** and **dhcpd6.conf** configuration files. You must provide following parameters for authentication while defining option space:

- Authentication code—The authentication code is either 0 or 1; where 0 indicates that authentication is not required, and 1 indicates that MD5 checksum is required.
- Client identifier—The client identifier must be 'exr-config'.
- MD5 checksum—This is chassis serial number. It can be obtained using **echo -n $SERIALNUMBER | md5sum | awk '{print $1}'**.

Here is the sample **dhcpd.conf** configuration. In the example below, the option space called **VendorInfo** is defined with three parameters for authentication:

```
class "vendor-classes" {
   match option vendor-class-identifier;
}

option space VendorInfo;
option VendorInfo.clientId code 1 = string;
option VendorInfo.authCode code 2 = unsigned integer 8;
option VendorInfo.md5sum code 3 = string
option vendor-specific code 43 = encapsulate VendorInfo;
subnet 10.65.2.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option routers 10.65.2.1;
  range 10.65.2.1 10.65.2.200;
}
host xrv9k-1-mgmt {
   hardware ethernet 00:50:60:45:67:01;
   fixed-address 10.65.2.39;
   vendor-option-space VendorInfo;
   option VendorInfo.clientId "exr-config";
   option VendorInfo.authCode 1;
   option VendorInfo.md5sum "aedf5c457c36390c664f5942ac1ae3829";
   option bootfile-name "http://10.65.2.1:8800/admin-cmd.sh";
}
```

Here is the sample **dhcpd6.conf** configuration file. In the example below, the option space called **VendorInfo** is defined that has code width 2 and length width 2 (as per dhcp standard for IPv6) with three parameters for authentication:

```
log-facility local7;
option dhcp6.name-servers 2001:1451:c632:1::1;
option dhcp6.domain-search "cisco.com";
dhcpv6-lease-file-name "/var/lib/dhcpd/dhcpd6.leases";
option dhcp6.info-refresh-time 21600;
option dhcp6.bootfile-url code 59 = string;
option dhcp6.user-class code 15 = string;
option space CISCO-EXR-CONFIG code width 2 length width 2;
option CISCO-EXR-CONFIG.client-identifier code 1 = string;
option CISCO-EXR-CONFIG.authCode code 2 = integer 8;
option CISCO-EXR-CONFIG.md5sum code 3 = string;
option vsio.CISCO-EXR-CONFIG code 9 = encapsulate CISCO-EXR-CONFIG;
```

```
subnet6 2001:1451:c632:1::/64{
 range6 2001:1451:c632:1::2 2001:1451:c632:1::9;
 #host NCS5501-2 {
      #host-identifier option dhcp6.client-id
00:02:00:00:00:09:46:4f:43:32:30:35:31:52:30:57:34:00;
       option CISCO-EXR-CONFIG.client-identifier "exr-config";
       option CISCO-EXR-CONFIG.authCode 1;
       #invalid md5
       #option CISCO-EXR-CONFIG.md5sum "90fd845ac82c77f834d57a034658d0f1";
       #valid md5
       option CISCO-EXR-CONFIG.md5sum "90fd845ac82c77f834d57a034658d0f0";
       if option dhcp6.user-class = 00:04:69:50:58:45 {
       option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/NCS5501-2/image.iso";
       }
       else {
        #option dhcp6.bootfile-url
"http://[2001:1851:c632:1::1]/NCS5501-2/ncs5500-mini-x.iso.sh";
        option dhcp6.bootfile-url "http://[2001:1851:c632:1::1]/NCS5501-2/ztp.cfg";
       }
 #}
}
```

ZTP Bootscript

If you want to hard code a script to be executed every boot, configure the following.

```
conf t
    ztp bootscript /disk0:/myscript
commit
```

The above configuration will wait for the first data-plane interface to be configured and then wait an additional minute for the management interface to be configured with an IP address, to ensure that we have connectivity in the third party namespace for applications to use. If the delay is not desired, use:

```
conf t
    ztp bootscript preip /disk0:/myscript
commit
```

Note When the above command is first configured, you will be prompted if you wish to invoke it now. The prompt helps with testing.

This is the example content of **/disk0:/myscript**:

```
#!/bin/bash
exec &> /dev/console # send logs to console
source /pkg/bin/ztp_helper.sh

# If we want to only run one time:
xrcmd "show running" | grep -q myhostname
if [[ $? -eq 0 ]]; then
    echo Already configured
fi

# Set the hostname
cat >/tmp/config <<%%
!! XR config example
```

```
hostname myhostname
%%
xrapply /tmp/config

#
# Force an invoke of ZTP again. If there was a username normally it would not run. This
forces it.
# Kill off ztp if it is running already and suppress errors to the console when ztp runs
below and
# cleans up xrcmd that invokes it. ztp will continue to run however.
#
xrcmd "ztp terminate noprompt" 2>/dev/null
xrcmd "ztp initiate noprompt" 2>/dev/null
```

ZTP Utilities

ZTP includes a set of shell utilities that can be sourced within the user script. **ztp_helper.sh** is a shell script that can be sourced by the user script. **ztp_helper.sh** provides simple utilities to access some XR functionalities. Following are the bash functions that can be invoked:

- **xrcmd**—Used to run a single XR exec command:

  ```
  xrcmd “show running”
  ```

- **xrapply**—Applies the block of configuration, specified in a file:

  ```
  cat >/tmp/config <<%%
  !! XR config example
  hostname node1-mgmt-via-xrapply
  %%
  xrapply /tmp/config
  ```

- **xrapply_with_reason**—Used to apply a block of XR configuration along with a reason for logging purpose:

  ```
  cat >/tmp/config <<%%
  !! XR config example
  hostname node1-mgmt-via-xrapply
  %%
  xrapply_with_reason "this is a system upgrade" /tmp/config
  ```

- **xrapply_string**—Used to apply a block of XR configuration in one line:

  ```
  xrapply_string "hostname foo\ninterface GigabitEthernet0/0/0/0\nipv4 address 1.2.3.44
  255.255.255.0\n"
  ```

- **xrapply_string_with_reason**—Used to apply a block of XR configuration in one line along with a reason for logging purposes:

  ```
  xrapply_string_with_reason ”system renamed again" "hostname venus\n interface
  TenGigE0/0/0/0\n ipv4 address 172.30.0.144/24\n”
  ```

- **xrreplace**—Used to apply XR configuration replace in XR namespace via a file.

  ```
  cat rtr.cfg <<%%
  !! XR config example
  hostname node1-mgmt-via-xrreplace
  ```

```
%%
xrreplace rtr.cfg
```

- **admincmd**—Used to run an admin CLI command in XR namespace. Logs can be found in **/disk0:/ztp/ztp_admincmd.log**

```
admincmd running [show platform]

ztp-user connected from 192.0168.0.1 using console on host
sysadmin-vm:0_RP0# show platform | nomore
Tue Jan 30 23:12:30.757 UTC
Location Card Type HW State SW State Config State
----------------------------------------------------------------------------
0/RP0 NCS-5501 OPERATIONAL OPERATIONAL NSHUT
0/FT0 NCS-1RU-FAN-FW OPERATIONAL N/A NSHUT
0/FT1 NCS-1RU-FAN-FW OPERATIONAL N/A NSHUT
0/PM0 NCS-1100W-ACFW OPERATIONAL N/A NSHUT
0/PM1 NCS-1100W-ACFW OPERATIONAL N/A NSHUT
```

- **xrapply_with_extra_auth**—Used to apply XR configuration that requires authentication, in XR namespace via a file. The **xrapply_with_extra_auth** API is used when configurations that require additional authentication to be applied such as alias, flex groups.

```
cat >/tmp/config <<%%
!! XR config example
alias exec alarms show alarms brief system active
alias exec version run cat /etc/show_version.txt
%%
xrapply_with_extra_auth >/tmp/config
```

- **xrreplace_with_extra_auth**—Used to apply XR configuration replace in XR namespace via a file The **xrreplace_with_extra_auth** API is used when configurations that require additional authentication to be applied such as alias, flex groups

```
cat >/tmp/config <<%%
!! XR config example
alias exec alarms show alarms brief system active
alias exec version run cat /etc/show_version.txt
%%
xrreplace_with_extra_auth >/tmp/config
```

Customize the ZTP Configurable Options

Starting with Cisco IOS XR Release 7.0.1, you can customize the following ZTP configurable options in the *ztp.ini* file:

- `ZTP`: You can enable or disable ZTP at boot using CLI or by editing the *ztp.ini* file.
- `Retry`: Set the ZTP DHCP retry mechanism: The available values are infinite and once.
- `Fetcher Priority`: You can modify the default priority of the Fetcher. Allowed range is from 0 to 9. Priority is in the increasing order.
- `progress_bar`: Enable Progress Bar on the console. By default, the progress bar is disabled. To enable the progress bar, add the following entry in the `ztp.ini` file.

```
    [Options]
    progress_bar: True
```

The following example shows the sample of the `ztp.ini` file:

```
[Startup]
start: True
retry_forever: True

[Fetcher Priority]
Mgmt4: 0
Mgmt6: 1
DPort4: 2
DPort6: 3
```

Enable ZTP Using CLI

If you want to enable ZTP using CLI, use the **ztp enable** command.

Configuration example

```
Router#ztp enable
Fri Jul 12 16:09:02.154 UTC
Enable ZTP? [confirm] [y/n] :y
ZTP Enabled.
```

Disable ZTP Using CLI

If you want to disable ZTP using CLI, use the **ztp disable** command.

Configuration example

```
Router#ztp disable
Fri Jul 12 16:07:18.491 UTC
Disable ZTP? [confirm] [y/n] :y
ZTP Disabled.
Run ZTP enable to run ZTP again.
```

Examples

ZTP logs its operation on the flash file system in the directory **/disk0:/ztp/**. ZTP logs all the transaction with the DHCP server and all the state transition.

The following example displays the execution of a simple configuration script downloaded from a data interface using the command **ztp initiate interface Ten 0/0/0/0 verbose**, this script will unshut all the interfaces of the system and configure a load interval of 30 seconds on all of them.

```
#!/bin/bash
##############################################################################
# *** Be careful this is powerful and can potentially destroy your system ***
#                *** !!! Use at your own risk !!! ***
#
# Script file should be saved on the backend HTTP server
##############################################################################

source ztp_helper.sh
config_file="/tmp/config.txt"
interfaces=$(xrcmd "show interfaces brief")

function activate_all_if(){
```

```
  arInt=($(echo $interfaces | grep -oE '(Te|Fo|Hu)[0-9]*/[0-9]*/[0-9]*/[0-9]*'))
  for int in ${arInt[*]}; do
    echo -ne "interface $int\n no shutdown\n load-interval 30\n" >> $config_file
  done
  xrapply_with_reason "Initial ZTP configuration" $config_file
}

### Script entry point
if [ -f $config_file ]; then
  /bin/rm -f $config_file
else
  /bin/touch $config_file
fi
activate_all_if;
exit 0
```

The following example displays the the console log of ztp initiate interface hundredGigE 0/1/0/4:

```
RP/0/RSP0/CPU0:vkg1#ztp initiate interface hundredGigE 0/1/0/4 verbose
Invoke ZTP? (this may change your configuration) [confirm] [y/n] :y
ZTP will now run in the background.
Please use "show logging" or look at /disk0:/ztp/ztp.log to check progress.
RP/0/RSP0/CPU0:vkg1#(Global VRF NS                 ) Fri Sep  1 12:47:46 UTC 2017: (pid
2984) (/pkg/bin/ztp.sh)                        : State change to IS_STARTING
(Global VRF NS                ) Fri Sep  1 12:47:49 UTC 2017: (pid  2984) (/pkg/bin/ztp.sh)
                        : Mgmt interface is brought up and ipv6 enabled
(Global VRF NS                ) Fri Sep  1 12:48:04 UTC 2017: (pid  2984) (/pkg/bin/ztp.sh)
                        : Final interface list: Hg0_1_0_4
(Global VRF NS                 ) Fri Sep  1 12:48:09 UTC 2017: (pid  4270)
(/pkg/bin/ztp_invoke_dhcp.sh)           : Starting Global VRF dhclient for: Hg0_1_0_4
(Global VRF NS                ) Fri Sep  1 12:48:14 UTC 2017: (pid  2984) (/pkg/bin/ztp.sh)
                        : ERROR: There is no gateway IP as the server is behind the gateway
(Global VRF NS                ) Fri Sep  1 12:48:34 UTC 2017: (pid  2984) (/pkg/bin/ztp.sh)
                        : Download finished. Waiting on config to be applied now.
(Global VRF NS                ) Fri Sep  1 12:49:00 UTC 2017: (pid  2984) (/pkg/bin/ztp.sh)
                        : ZTP is applying config
(Global VRF NS                ) Fri Sep  1 12:49:13 UTC 2017: (pid  2984) (/pkg/bin/ztp.sh)
                        : Exiting SUCCESSFULLY
```