



System Management Configuration Guide for Cisco NCS 560 Series Routers, IOS XR Release 7.11.x

First Published: 2023-12-01

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

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

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

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

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

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

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

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

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

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

© 2023 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1	YANG Data Models for System Management Features 1	
	Using YANG Data Models 1	
CHAPTER 2	Configuring Flexible Command Line Interface 3	
	Flexible CLI Configuration Groups 3	
	Flexible Configuration Restrictions 3	
	Configuring a Configuration Group 5	
	Simple Configuration Group: Example 6	
	Configuration Group Applied to Different Places: Example 7	
	Verifying the Configuration of Configuration Groups 7	
	Regular Expressions in Configuration Groups 8	
	Configuration Examples Using Regular Expressions 15	
	Configuration Group with Regular Expression: Example 15	
	Configuration Group Inheritance with Regular Expressions: Example 17	
	Layer 2 Transport Configuration Group: Example 18	
	Configuration Group Precedence: Example 19	
	Changes to Configuration Group are Automatically Inherited: Example 19	
	Configuration Examples for Flexible CLI Configuration 20	
	Basic Flexible CLI Configuration: Example 20	
	Interface MTU Settings for Different Interface Types: Example 21	
	ACL Referencing: Example 23	
	ISIS Hierarchical Configuration: Example 24	
	OSPF Hierarchy: Example 28	
	Link Rundling Usage: Example 31	

CHAPTER 3 Configuring Manageability 33

CHAPTER 4

CHAPTER 5

Information about XML Manageability How to Configure Manageability 33 Configuring the XML Agent 33 Configuration Examples for Manageability 34 Enabling VRF on an XML Agent: Example 34 Configuring Object Tracking 37 Configuring Object Tracking Prerequisites for Implementing Object Tracking 37 Information about Object Tracking How to Implement Object Tracking 38 Tracking the Line Protocol State of an Interface 38 Tracking IP Route Reachability 40 Building a Track Based on a List of Objects 41 Building a Track Based on a List of Objects - Threshold Percentage Building a Track Based on a List of Objects - Threshold Weight 45 Tracking IPSLA Reachability 46 Configuration Examples for Configuring Object Tracking 47 Additional References 50 Configuring Physical and Virtual Terminals 53 Prerequisites for Implementing Physical and Virtual Terminals 53 Information About Implementing Physical and Virtual Terminals 53 Line Templates Line Template Configuration Mode Line Template Guidelines 54 **Terminal Identification** vtv Pools 55 How to Implement Physical and Virtual Terminals on Cisco IOS XR Software 56 **Modifying Templates** 56 Creating and Modifying vty Pools Monitoring Terminals and Terminal Sessions Configuration Examples for Implementing Physical and Virtual Terminals

CHAPTER 6 Configuring Simple Network Management Protocol 63 Prerequisites for Implementing SNMP 63 Restrictions for SNMP use on Cisco IOS XR Software 63 Information about Implementing SNMP 64 SNMP Functional Overview **64 SNMP Manager** SNMP Agent 64 **MIB** 64 **SNMP Versions** Comparison of SNMPv1, v2c, and v3 66 Security Models and Levels for SNMPv1, v2, v3 66 SNMPv3 Benefits **67** SNMPv3 Costs 68 User-Based Security Model View-Based Access Control Model IP Precedence and DSCP Support for SNMP Session MIB support on subscriber sessions SNMP Notifications **69** Session Types 71 How to Implement SNMP on Cisco IOS XR Software 71 Configuring SNMPv3 71 Configuring SNMPv3: Examples Configuring SNMP Trap Notifications 77 Configuring Trap Notifications: Example 78 Setting the Contact, Location, and Serial Number of the SNMP Agent Defining the Maximum SNMP Agent Packet Size Changing Notification Operation Values Setting IP Precedence and DSCP Values 82 Setting an IP Precedence Value for SNMP Traffic: Example 83 Setting an IP DSCP Value for SNMP Traffic: Example 83 Displaying SNMP Context Mapping Monitoring Packet Loss 84 Configuring MIB Data to be Persistent 85

CHAPTER 7 Configuring Periodic MIB Data Collection and Transfer 89 Prerequisites for Periodic MIB Data Collection and Transfer 89

Information About Periodic MIB Data Collection and Transfer 89

SNMP Objects and Instances 89

Bulk Statistics Object Lists 90

Bulk Statistics Schemas 90

Bulk Statistics Transfer Options 90

Benefits of Periodic MIB Data Collection and Transfer 90

How to Configure Periodic MIB Data Collection and Transfer 91

Configuring a Bulk Statistics Object List 91

Configuring a Bulk Statistics Schema 92

Configuring Bulk Statistics Transfer Options 93

Periodic MIB Data Collection and Transfer: Example 9

CHAPTER 8 Configuring Cisco Discovery Protocol 99

Prerequisites for Implementing CDP 99

Information About Implementing CDP 99

How to Implement CDP on Cisco IOS XR Software 101

Enabling CDP 101

Modifying CDP Default Settings 102

Monitoring CDP 103

Configuration Examples for Implementing CDP 104

CHAPTER 9 What is Smart Licensing? 107

What is Flexible Consumption Model? 108

Software Innovation Access 111

Verification 112

How Does Smart Licensing Work? 114

What is Cisco Smart Software Manager? 116

Smart Licensing Deployment Options 117

YANG Data Models for Smart Licensing 118

Configuring Smart Licensing 119

```
Prerequisites for Configuring Smart Licensing 119
        Setting up the Router for Smart Licensing 120
        Configuring a Communications Connection Between the Router and Cisco Smart Software
          Manager 120
          Configuring a Direct Cloud Connection 120
          Configuring a Connection Through an HTTP Proxy 122
          Connecting to CSSM On-Premise 124
          Installing CSSM On-Premise 126
     Registering and Activating Your Router
       Generating a New Token from CSSM
        Registering Your Device With the Token 130
        Renewing Your Smart Licensing Registration
        Deregistering Your Router from CSSM 131
     Verifying the Smart Licensing Configuration 131
       Smart Licensing Configuration Examples 133
          Example: Viewing the Call Home Profile 133
          Example: Viewing License Information Before Registration
          Example: Registering the Router
          Example: Viewing License Information After Registration 137
Upgrading Field-Programmable Device 141
     Prerequisites for FPD Image Upgrades 141
     Overview of FPD Image Upgrade Support 141
     FPD upgrade service 142
        Determining Upgrade Requirement 142
        Manual FPD upgrade 143
         How to Upgrade FPD Images 143
          Configuration Examples for FPD Image Upgrade 144
        Auto FPD Upgrade 147
```

Limitations and Usage Guidelines 147

Configuring Auto FPD During ISSU 153
Automatic FPD Upgrade for PSU 162

Upgrade Failure 162

Configuring Auto FPD During System Upgrade 149

CHAPTER 10

vii

Contents



YANG Data Models for System Management Features

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

• Using YANG Data Models, on page 1

Using YANG Data Models

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

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

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

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

Using YANG Data Models



Configuring Flexible Command Line Interface

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

- Flexible CLI Configuration Groups, on page 3
- Flexible Configuration Restrictions, on page 3
- Configuring a Configuration Group, on page 5
- Verifying the Configuration of Configuration Groups, on page 7
- Regular Expressions in Configuration Groups, on page 8
- Configuration Examples for Flexible CLI Configuration, on page 20

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.*|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].
```

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 O/RPO/CPUO
```

^{&#}x27;TenGigE.*|HundredGigE.*—To match on either TenGigE.* or HundredGigE.*.

• 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 gigabitethernet0/0/0/* ---- where * is 0 to 31
apply-group G-INTERFACE
```

This configuration is not permitted because it cannot be determined whether the interface GigabitEthernet0/0/0/* configuration inherits mtu 1500 or mtu 2000. Both expressions in the configuration group match GigabitEthernet0/0/0/*.

• Up to eight configuration groups are permitted on one apply-group command.

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.

Procedure

Step 1 configure

Example:

RP/0/RP0/CPU0:router# configure

Enters mode.

Step 2 group group-name

Example:

```
RP/0/RP0/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.

Step 3 Enter configuration commands, starting from global configuration mode. Use regular expressions for interface names and other variable instances.

Example:

```
RP/0/RP0/CPU0:router(config) # group g-interf
RP/0/RP0/CPU0:router(config-GRP) # interface 'GigabitEthernet.*'
RP/0/RP0/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 Configuration Group Inheritance with Regular Expressions: Example, on page 17. This example is applicable to all Gigabit Ethernet interfaces.

Step 4 end-group

Example:

```
RP/0/RP0/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:

```
RP/0/RP0/CPU0:router(config) # interface GigabitEthernet0/2/0/0
RP/0/RP0/CPU0:router(config-if) # apply-group g-interf
```

Adds the configuration of the configuration group into the router configuration applicable at the location that the group 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 GigabitEthernet0/2/0/0GigabitEthernet0/0/0/0. 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/RP0/CPU0:router(config) # group g-logging
RP/0/RP0/CPU0:router(config-GRP) # logging trap notifications
RP/0/RP0/CPU0:router(config-GRP) # logging console debugging
RP/0/RP0/CPU0:router(config-GRP) # logging monitor debugging
RP/0/RP0/CPU0:router(config-GRP) # logging buffered 10000000
RP/0/RP0/CPU0:router(config-GRP) # end-group
RP/0/RP0/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/RP0/CPU0:router(config) # group g-interfaces
RP/0/RP0/CPU0:router(config-GRP) # interface 'TenGigE.*'
RP/0/RP0/CPU0:router(config-GRP-if) # mtu 1500
RP/0/RP0/CPU0:router(config-GRP-if) # exit
RP/0/RP0/CPU0:router(config-GRP) # interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-if) # mtu 1000
RP/0/RP0/CPU0:router(config-GRP-if) # exit
RP/0/RP0/CPU0:router(config-GRP) # interface 'HundredGigE.*'
RP/0/RP0/CPU0:router(config-GRP-if) # mtu 2000
RP/0/RP0/CPU0:router(config-GRP-if) # end-group
```

This group can be applied to Gigabit Ethernet, TenGigE and HundredGigE interface and in each instance the applicable MTU is applied. For instance, in the following example, the Gigabit Ethernet interface is configured to have an MTU of 1000:

```
RP/0/RP0/CPU0:router(config) # interface GigabitEthernet0/2/0/0
RP/0/RP0/CPU0:router(config-if) # apply-group g-interfaces
RP/0/RP0/CPU0:router(config-if) # ipv4 address 2.2.2.2 255.255.255.0
```

In the following example, the TenGigE interface is configured to have an MTU of 1500:

```
RP/0/RP0/CPU0:router(config) # interface TenGigE0/0/0/16
RP/0/RP0/CPU0:router(config-if) # apply-group g-interfaces
RP/0/RP0/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:

Procedure

	Command or Action	Purpose
Step 1	show running-config group [group-name]	Displays the contents of all or a specific configured configuration group.
	Example:	
	RP/0/RP0/CPU0:router# show running-config group	
	<pre>group g-int-ge interface 'GigabitEthernet.*' mtu 1000 negotiation auto ! end-group</pre>	
Step 2	<pre>show running-config Example: show running-config Example:</pre>	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

	Command or Action	Purpose
	RP/0/RP0/CPU0:router# show running-config interface group G-INTERFACE-MTU interface 'GigabitEthernet.*' mtu 1500 ! end-group interface GigabitEthernet0/4/1/1 apply-group G-INTERFACE-MTU ! interface GigabitEthernet0/0/0/1 apply-group group G-INTERFACE-MTU mtu 2000	applied to interface GigabitEthernet0/0/0/1, 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.
Step 3	show running-config inheritance	Displays the inherited configuration wherever
	Example:	a configuration group has been applied.
	<pre>RP/0/RP0/CPU0:router# show running-config inheritance group G-INTERFACE-MTU interface 'GigabitEthernet.*' mtu 1500 ! end-group .</pre>	
	<pre>interface GigabitEthernet0/4/1/1 ## Inherited from group G-INTERFACE-MTU mtu 1500 ! interface GigabitEthernet0/0/0/1 mtu 2000 !</pre>	
Step 4	show running-config interface x/y/z inheritance detail	Displays the inherited configuration for a specific configuration command.
	Example: RP/0/RP0/CPU0:router# show running-config interface GigabitEthernet0/4/1/1 inheritance detailExample: interface GigabitEthernet0/4/1/1 ## Inherited from group G-INTERFACE-MTU mtu 1500	

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.



Note

Not all POSIX regular expressions are supported.

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/RP0/CPU0:router(config-GRP)# interface ?
```

```
'RegExp': ATM Network Interface(s)
ΑТМ
BVI
                 'RegExp': Bridge-Group Virtual Interface
                 'RegExp': Aggregated Ethernet interface(s)
Bundle-Ether
GigabitEthernet
                'RegExp': GigabitEthernet/IEEE 802.3 interface(s)
IMA
                 'RegExp': ATM Network Interface(s)
                 'RegExp': Loopback interface(s)
Loopback
                 'RegExp': Ethernet/IEEE 802.3 interface(s)
MgmtEth
Multilink
                'RegExp': Multilink network interface(s)
Null
                 'RegExp': Null interface
PW-Ether
                 'RegExp': PWHE Ethernet Interface
PW-TW
                 'RegExp': PWHE VC11 IP Interworking Interface
                 'RegExp': Serial network interface(s)
Serial
                 'RegExp': GRE/IPinIP Tunnel Interface(s)
tunnel-ip
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-12t
   interface 'Gi.*\..*' 12transport
.
.
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//CPU0:router(config) #group FB flexi snmp
RP/0//CPU0:router(config-GRP) # snmp-server vrf '.*'
RP/0//CPU0:router(config-GRP-snmp-vrf) # host 1.1.1.1 traps version 2c group 1
RP/0//CPU0:router(config-GRP-snmp-vrf)# host 1.1.1.1 informs version 2c group 1
RP/0//CPU0:router(config-GRP-snmp-vrf) # context group 1
RP/0//CPU0:router(config-GRP-snmp-vrf)#
RP/0//CPU0:router(config-GRP-snmp-vrf)#commit
RP/0//CPU0:router(config-GRP-snmp-vrf) #root
RP/0//CPU0:router(config)#
RP/0//CPU0:router(config) #snmp-server vrf vrf1
RP/0//CPU0:router(config-snmp-vrf)#snmp-server vrf vrf10
RP/0//CPU0:router(config-snmp-vrf)#!
RP/0//CPU0:router(config-snmp-vrf)#snmp-server vrf vrf100
RP/0//CPU0:router(config-snmp-vrf)#
RP/0//CPU0:router(config-snmp-vrf)#commit
RP/0//CPU0:router(config-snmp-vrf)#root
```

```
RP/0//CPU0:router(config)#
RP/0//CPU0:router(config)#apply-group FB_flexi_snmp
RP/0//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//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 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 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 GigabitEthernet0/0/0/0
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 GigabitEthernet0/0/0/0- 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/RP0/CPU0:router(config) # group g-isis-gige
RP/0/RP0/CPU0:router(config-GRP) # router isis '.*'
RP/0/RP0/CPU0:router(config-GRP-isis) # interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-isis-if) # lsp-interval 20
RP/0/RP0/CPU0:router(config-GRP-isis-if) # hello-interval 40
RP/0/RP0/CPU0:router(config-GRP-isis-if) # address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-GRP-isis-if-af) # metric 10
RP/0/RP0/CPU0:router(config-GRP-isis-if-af) # end-group
RP/0/RP0/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:

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/RP0/CPU0:router(config) # group g-ospf
RP/0/RP0/CPU0:router(config-GRP) # router ospf '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf) # nsf cisco
RP/0/RP0/CPU0:router(config-GRP-ospf) # packet-size 3000
RP/0/RP0/CPU0:router(config-GRP-ospf) # end-group
RP/0/RP0/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/RP0/CPU0:router(config) # group g-ospf
RP/0/RP0/CPU0:router(config-GRP) # router ospf '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf) # area '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf-ar) # packet-size 2000
RP/0/RP0/CPU0:router(config-GRP-ospf) # end-group
RP/0/RP0/CPU0:router(config) # apply-group g-ospf
RP/0/RP0/CPU0:router(config) # router ospf 200
RP/0/RP0/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/RP0/CPU0:router(config) # group g-l2trans-if
RP/0/RP0/CPU0:router(config-GRP) # interface 'TenGigE.*\..*' l2transport
RP/0/RP0/CPU0:router(config-GRP) # mtu 1514
RP/0/RP0/CPU0:router(config-GRP) # end-group

RP/0/RP0/CPU0:router(config) # interface TenGigE0/0/0/0.1 l2transport
RP/0/RP0/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/RP0/CPU0:router(config) # group g-ospf2
RP/0/RP0/CPU0:router(config-GRP) # router ospf '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf) # area '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf-ar) # cost 2
RP/0/RP0/CPU0:router(config-GRP-ospf-ar) # end-group
RP/0/RP0/CPU0:router(config) # group g-ospf100
RP/0/RP0/CPU0:router(config-GRP) # router ospf '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf) # area '.*'
RP/0/RP0/CPU0:router(config-GRP-ospf-ar) # cost 100
RP/0/RP0/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/RP0/CPU0:router(config)# router ospf 0
RP/0/RP0/CPU0:router(config-ospf)# apply-group g-ospf100
RP/0/RP0/CPU0:router(config-ospf)# area 0
RP/0/RP0/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 'GigabitEthernet.*'
  mtu 1500
  !
end-group
interface POSO/0/0/0
  apply-group g-interface-mtu
!
```

Now you change the configuration group as in this example:

```
RP/0/RP0/CPU0:router(config) # group g-interface-mtu
RP/0/RP0/CPU0:router(config-GRP) # interface 'GigabitEthernet.*'
RP/0/RP0/CPU0:router(config-GRP-if) # mtu 2000
```

```
RP/0/RP0/CPU0:router(config-GRP-if) # end-group
```

When this configuration group is committed, the MTU configuration for interface GigabitEthernet0/0/0/0 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/RP0/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/RP0/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/RP0/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/RP0/CPU0:router# show running-config 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/RP0/CPU0:router# show mac-accounting 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/RP0/CPU0:router# show running group 12tr
group 12tr
interface 'GigabitEthernet0/0/0/3.*'
mtu 1500
!
interface 'GigabitEthernet0/0/0/9\..*'
mtu 1400
!
interface 'GigabitEthernet0/0/0/9\..*' 12transport
mtu 1400
!
end-group

RP/0/RP0/CPU0:router# show running | inc apply-group
Building configuration...
apply-group 12tr
```

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

```
RP/0/RP0/CPU0:router# show running interface gigabitEthernet0/0/0/30
interface GigabitEthernet0/0/0/30
RP/0/RP0/CPU0:router# show running interface gigabitEthernet0/0/0/30 inheritance detail
interface GigabitEthernet0/0/0/30
## Inherited from group 12tr
mtu 1500
RP/0/RP0/CPU0:router# show running interface gigabitEthernet0/0/0/9.800
interface GigabitEthernet0/0/0/9.800
  encapsulation dot1q 800
RP/0/RP0/CPU0:router# show running interface gigabitEthernet0/0/0/9.800 inheritance
detail
interface GigabitEthernet0/0/0/9.800
## Inherited from group 12tr
mt.u 1400
encapsulation dot1q800
RP/0/RP0/CPU0:router# show running interface gigabitEthernet0/0/0/9.250
interface GigabitEthernet0/0/0/9.250 12transport
  encapsulation dot1q 250
RP/0/RP0/CPU0:router# show running interface gigabitEthernet0/0/0/9.800 inheritance
detail
interface GigabitEthernet0/0/0/9.250 12transport
encapsulation dot1q250
## Inherited from group 12tr
mtu 1400
```

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

```
RP/0/RP0/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
     O packets input, O bytes, O total input drops
     O 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
     O packets output, O bytes, O total output drops
     Output O broadcast packets, O multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     O output buffer failures, O output buffers swapped out
RP/0/RP0/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
     O drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     O packets output, O bytes, O total output drops
     Output 0 broadcast packets, 0 multicast packets
RP/0/RP0/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
 Laver 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/RP0/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/RP0/CPU0:router# show running | inc apply-group
Building configuration...
apply-group isis 12tr isis2 mpp bundle1 acref
```

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

```
RP/0/RP0/CPU0:router# show running interface gigabitEthernet 0/0/0/30
interface GigabitEthernet0/0/0/30
!
RP/0/RP0/CPU0:router# show running interface GigabitEthernet 0/0/0/30 inheritance detail
interface GigabitEthernet0/0/0/30
## Inherited from group 12tr
mtii 1500
## Inherited from group acref
ipv4 access-group adem ingress
## Inherited from group acref
ipv4 access-group adem egress
RP/0/RP0/CPU0:router# show running interface gigabitEthernet 0/0/0/31
interface GigabitEthernet0/0/0/31
RP/0/RP0/CPU0:router# show running interface GigabitEthernet 0/0/0/31 inheritance detail
interface GigabitEthernet0/0/0/31
## Inherited from group 12tr
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.

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/RP0/CPU0:router# show running router isis
router isis vink
```

```
net 49.0011.2222.2222.200
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
1
interface Bundle-Ether2
interface Loopback0
interface TenGigE0/0/0/0.3521
address-family ipv4 unicast
interface TenGigE0/0/0/0.3522
address-family ipv4 unicast
!
interface TenGigE0/0/0/0.3523
address-family ipv4 unicast
!
interface TenGigE0/0/0/0.3524
address-family ipv4 unicast
!
!
interface TenGigE0/0/0/0.3525
address-family ipv4 unicast
.
interface TenGigE0/0/0/0.3526
interface TenGigE0/0/0/0.3527
interface TenGigE0/0/0/0.3528
interface TenGigE0/0/0/1
address-family ipv4 unicast
!
```

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

```
RP/0/RP0/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
  1
 !
end-group
RP/0/RP0/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
   .
  !
 1
end-group
RP/0/RP0/CPU0:router# show running | inc apply-group
Building configuration...
apply-group isis 12tr isis2 mpp bundle1 acref
```

3. Check the inheritance view of the ISIS configuration:

```
router isis vink

net 49.0011.2222.2222.200

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
```

RP/0/RP0/CPU0:router# show running router isis inheritance detail

```
!
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 TenGigE0/0/0/0.3521
## 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 TenGigE0/0/0/0.3522
## 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 TenGigE0/0/0/0.3523
## 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 TenGigE0/0/0/0.3524
## 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 TenGigE0/0/0/0.3525
## Inherited from group isis
1sp-interval 40
 ## Inherited from group isis
hello-interval 15
address-family ipv4 unicast
 ## Inherited from group isis
 metric 50
interface TenGigE0/0/0/0.3525
## 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 TenGigE0/0/0/0.3527
## 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 TenGigE0/0/0/0.3528
## Inherited from group isis
1sp-interval 40
## Inherited from group isis
hello-interval 15
## Inherited from group isis
address-family ipv4 unicast
 ## Inherited from group isis
 metric 50
interface TenGigE0/0/0/1
## Inherited from group isis
1sp-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/RP0/CPU0:router# show isis interface TenGigE0/0/0/0.3528 | 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/RP0/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
```

```
\verb|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 TenGigE0/0/0/0.3891
  interface TenGigE0/0/0/0.3892
  .
  interface TenGigE0/0/0/0.3893
 interface TenGigE0/0/0/0.3894
 !
1
router ospf 100
router ospf 1000
router ospf 1001
```

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

```
RP/0/RP0/CPU0:router# show running group go-a
group go-a
router ospf '.*'
 area '.*'
  interface 'Gig.*'
   cost 200
  !
  - 1
end-group
RP/0/RP0/CPU0:router# show running group go-b
group go-b
router ospf '.*'
 area '.*'
  interface 'Gig.*'
   cost 250
   - 1
end-group
RP/0/RP0/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/RP0/CPU0:router# show running router ospf 1 inheritance detail 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 << apply-group in lowest submode gets highest priority cost 200 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 TenGigE0/0/0/0.3891

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

interface TenGigE0/0/0/0.3892
!
interface TenGigE0/0/0/0.3893
!
interface TenGigE0/0/0/0.3894

!

```
RP/0/RP0/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/RP0/CPU0:router# show running group bundle1
group bundle1
interface 'GigabitEthernet0/1/0/1[1-6]'
bundle id 1 mode active
!
end-group

RP/0/RP0/CPU0:router# show running | inc apply-group

Building configuration...
apply-group isis 12tr isis2 mpp bundle1
```

2. Check the local configuration:

```
RP/0/RP0/CPU0:router# show running interface gigabitEthernet 0/1/0/11
interface GigabitEthernet0/1/0/11
!

RP/0/RP0/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/RP0/CPU0:router# show running interface GigabitEthernet 0/1/0/11 inheritance detail
interface GigabitEthernet0/1/0/11
## Inherited from group bundle1
bundle id 1 mode active
!
```

4. Check that the inheritance configuration took effect:

RP/0/RP0/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 Full-duplex 1000Mb/s GigabitEthernet0/1/0/12 Active GigabitEthernet0/1/0/13 Full-duplex 1000Mb/s Active GigabitEthernet0/1/0/14 Full-duplex 1000Mb/s Active Full-duplex 1000Mb/s GigabitEthernet0/1/0/15 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 O 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 O output buffer failures, O output buffers swapped out O carrier transitions



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.
- Information about XML Manageability, on page 33
- How to Configure Manageability, on page 33
- Configuration Examples for Manageability, on page 34

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

This explains how to configure the XML agent.

Procedure

	Command or Action	Purpose
Step 1	<pre>xml agent [ssl] Example: RP/0/RP0/CPU0:router(config) # xml agent ssl</pre>	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	<pre>iteration on size iteration-size Example: RP/0/RP0/CPU0:router(config-xml-agent) # iteration on size 500</pre>	Configures the iteration size for large XML agent responses in KBytes. The default is 48.
Step 3	<pre>session timeout timeout Example: RP/0/RP0/CPU0:router(config-xml-agent) # session timeout 5</pre>	Configures an idle timeout for the XML agent in minutes. By default, there is no timeout.
Step 4	<pre>throttle { memory size process-rate tags } Example: RP/0/RP0/CPU0:router(config-xml-agent) # throttle memory 300</pre>	Configures the XML agent processing capabilities. • Specify the memory size in Mbytes. 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	<pre>vrf { vrfname default } [ipv4 access-list access-list-name] Example: RP/0/RP0/CPU0:router(config-xml-agent) # vrf vrf1</pre>	Configures the dedicated agent or SSL agent to receive and send messages via the specified VPN routing and forwarding (VRF) instance.

Configuration Examples for Manageability

Enabling VRF on an XML Agent: Example

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

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

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

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

Enabling VRF on an XML Agent: Example



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 **Additional References** section. To locate documentation for other commands that might appear in the course of performing a configuration task, see **Technical Documentation** section in the Additional References topic.

- Configuring Object Tracking, on page 37
- Prerequisites for Implementing Object Tracking, on page 37
- Information about Object Tracking, on page 38
- How to Implement Object Tracking, on page 38
- Configuration Examples for Configuring Object Tracking, on page 47
- Additional References, on page 50

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 **Additional References** section. To locate documentation for other commands that might appear in the course of performing a configuration task, see **Technical Documentation** section in the Additional References topic.

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.



Note

Object Tracking is an optional package. You must check if this package is installed on your system by running the command **show install active summary**.

Information about Object Tracking

Object tracking is a mechanism to track an object and to take an action on another object with no relationship to the tracked objects, based on changes to the properties of the object being tracked.

Each tracked object is identified by a unique name specified on the tracking command-line interface (CLI). Cisco IOS XR processes then use this name to track a specific object.

The tracking process periodically polls the tracked object and reports any changes to its state in terms of its being up or down, either immediately or after a delay, as configured by the user.

Multiple objects can also be tracked by means of a list, using 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 defined within a subset must be in an up state, so that the tracked object can also be in the up state.
- **Boolean OR function**—When the tracked list has been assigned a Boolean OR function, it means that at least one object defined within a subset must also be in an up state, so that the tracked object can also be in the up state.

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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	track track-name	Enters track configuration mode.
	Example:	• <i>track-name</i> —Specifies a name for the object to be tracked.
	<pre>RP/0/RP0/CPU0:router(config) # track track1</pre>	
Step 3	type line-protocol state	Creates a track based on the line protocol of ar
	Example:	interface.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-track)# type line-protocol state	
Step 4	<pre>interface type interface-path-id Example: RP/0/RP0/CPU0:router(config-track-line-prot)# interface atm 0/2/0/0.1</pre>	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	<pre>exit Example: RP/0/RP0/CPU0:router(config-track-line-prot)# exit</pre>	Exits the track line protocol configuration mode.
Step 6	<pre>(Optional) delay { up seconds down seconds } Example: RP/0/RP0/CPU0:router(config-track) # delay up 10</pre>	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/RP0/CPU0:router(config-track) # end or RP/0/RP0/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.

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.

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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	track track-name	Enters track configuration mode.
	Example:	 track-name—Specifies a name for the object to be tracked.
	<pre>RP/0/RP0/CPU0:router(config)# track track1</pre>	
Step 3	type route reachability	Configures the routing process to notify the
	Example:	tracking process when the state of the route changes due to a routing update.
	<pre>RP/0/RP0/CPU0:router(config-track) # type route reachability vrf internet</pre>	
Step 4	Use one of the following commands:	Configures the type of IP route to be tracked,
	• vrf vrf-table-name	which can consist of either of the following,
	• route ipv4 IP-prefix/mask	depending on your router type:
	Example:	 vrf-table-name—A VRF table name. IP-prefix/mask—An IP prefix consisting
	<pre>RP/0/RP0/CPU0:router(config-track-route)# vrf vrf-table-4</pre>	of the network and subnet mask (for example, 10.56.8.10/16).
	or	

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-track-route)# route ipv4 10.56.8.10/16	
Step 5	exit	Exits the track line protocol configuration mode.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config-track-line-prot)# exit</pre>	
Step 6	(Optional) delay { up seconds down seconds }	Schedules the delay that can occur between tracking whether the object is up or down.
	Example:	
	RP/0/RP0/CPU0:router(config-track)# delay up 10	
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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	track track-name	Enters track configuration mode.
	Example:	• <i>track-name</i> —Specifies a name for the object to be tracked.
	RP/0/RP0/CPU0:router(config)# track track1	
Step 3	type list boolean { and or } Example:	Configures a Boolean list object and enters track list configuration mode.
	RP/0/RP0/CPU0:router(config-track)# type list boolean and	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]	Specifies the object to be tracked by the list
	Example:	• <i>obect-name</i> —Name of the object to track.
	<pre>RP/0/RP0/CPU0:router(config-track-list)# object 3 not</pre>	• not—Negates the state of the object.
Step 5	exit	Exits the track line protocol configuration mode.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config-track-line-prot)# exit</pre>	
Step 6	(Optional) delay { up seconds down seconds }	Schedules the delay that can occur between tracking whether the object is up or down.
	Example:	
	RP/0/RP0/CPU0:router(config-track)# delay up 10	,

	Command or Action	Purpose
Step 7	Use one of the following commands:	Saves configuration changes.
	• end • commit	• When you issue the end command, the system prompts you to commit changes:
	<pre>Example: RP/0/RP0/CPU0:router(config-track) # end or RP/0/RP0/CPU0:router(config-track) # commit</pre>	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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	track track-name	Enters track configuration mode.
	Example:	 track-name—Specifies a name for the object to be tracked.
	<pre>RP/0/RP0/CPU0:router(config)# track track1</pre>	

	Command or Action	Purpose
Step 3	type list threshold percentage Example:	Configures a track of type threshold percentage list.
	RP/0/RP0/CPU0:router(config-track) # type list threshold percentage	
Step 4	object object-name Example:	Configures object 1, object 2, object 3 and object 4 as members of track type track1.
	<pre>RP/0/RP0/CPU0:router(config-track-list-threshold)# object 1 RP/0/RP0/CPU0:router(config-track-list-threshold)# object 2 RP/0/RP0/CPU0:router(config-track-list-threshold)# object 3 RP/0/RP0/CPU0:router(config-track-list-threshold)# object 4</pre>	
Step 5	threshold percentage up percentage down percentage Example:	Configures the percentage of objects that need to be UP or DOWN for the list to be considered UP or Down respectively.
	RP/0/RP0/CPU0:router(config-track-list-threshold)# threshold percentage up 50 down 33	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:	Saves configuration changes.
	• end • commit	• When you issue the end command, the system prompts you to commit changes:
	Example: RP/0/RP0/CPU0:router(config-track)# end	Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
	<pre>Or RP/0/RP0/CPU0:router(config-track)# commit</pre>	• 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.

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

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

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

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	track track-name	Enters track configuration mode.
	Example:	• <i>track-name</i> —Specifies a name for the object to be tracked.
	<pre>RP/0/RP0/CPU0:router(config) # track track1</pre>	
Step 3	type list threshold weight	Configures a a track of type, threshold weighted
	Example:	list.
	RP/0/RP0/CPU0:router(config-track)# type list threshold weight	
Step 4	object object-name weight weight Example:	Configures object 1, object 2 and object 3 as members of track t1 and with weights 10, 5 and 3 respectively.
	RP/0/RP0/CPU0:router(config-track-list-threshold)# object 1 weight 10 RP/0/RP0/CPU0:router(config-track-list-threshold)# object 2 weight 5 RP/0/RP0/CPU0:router(config-track-list-threshold)# object 3 weight 3	
Step 5	threshold weight up weight down weight	Configures the range of weights for the objects
	Example:	that need to be UP or DOWN for the list to be considered UP or DOWN respectively. In this
	RP/0/RP0/CPU0:router(config-track-list-threshold)# threshold weight up 10 down 5	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).

	Command or Action	Purpose
Step 6	Use one of the following commands:	Saves configuration changes.
	• end • commit	• When you issue the end command, the system prompts you to commit changes:
	<pre>Example: RP/0/RP0/CPU0:router(config-track)# end or RP/0/RP0/CPU0:router(config-track)# commit</pre>	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.

	Command or Action	Purpo	se
Step 1	configure	Enters	s global configuration mode.
	Example:		
0, 0	RP/0/RP0/CPU0:router# configure	Б.	. 1 "
Step 2	track track-name	Enters	s track configuration mode.
	Example:	Note	Special characters are not allowed in a <i>track-name</i> .
	RP/0/RP0/CPU0:router(config)# track t1		ruck name.

	Command or Action	Purpose
Step 3	<pre>type rtr ipsla-no reachability Example: RP/0/RP0/CPU0:router(config-track) # type rtr 100 reachability</pre>	Specifies the IP SLA operation ID to be tracked for reachability. Values for the <i>ipsla-no</i> can range from 1 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/RP0/CPU0:router(config) # track track1
RP/0/RP0/CPU0:router(config-track) # type rtr 1 reachability
RP/0/RP0/CPU0:router(config-track) # delay up 5
RP/0/RP0/CPU0:router(config-track) # delay down 10
```

Configuration Examples for Configuring Object Tracking

Tracking Whether the Interface Is Up or Down: Running Configuration Example

```
track connection100
  type list boolean and
  object object3 not
  delay up 10
  !
interface service-ipsec 23
  line-protocol track connection100
```

Tracking the Line Protocol State of an Interface: Running Configuration Example

In this example, traffic arrives from interface service-ipsec1 and exits through interface gigabitethernet0/0/0/3:

```
track IPSec1
  type line-protocol state
    interface gigabitethernet0/0/0/3
  !
interface service-ipsec 1
  ipv4 address 70.0.0.1 255.255.255.0
  profile vrf1_profile_ipsec
  line-protocol track IPSec1
  tunnel source 80.0.0.1
  tunnel destination 80.0.0.2
  service-location preferred-active 0/0/1
!
```

This example displays the output from the **show track** command after performing the previous example:

Tracking IP Route Reachability: Running Configuration Example

In this example, traffic arriving from interface service-ipsec1 has its destination in network 7.0.0.0/24. This tracking procedure follows the state of the routing protocol prefix to signal when there are changes in the routing table.

```
track PREFIX1
  type route reachability
   route ipv4 7.0.0.0/24
  !
  interface service-ipsec 1
  vrf 1
  ipv4 address 70.0.0.2 255.255.255.0
  profile vrf_1_ipsec
  line-protocol track PREFIX1
  tunnel source 80.0.0.2
  tunnel destination 80.0.0.1
  service-location preferred-active 0/2/0
```

Building a Track Based on a List of Objects: Running Configuration Example

In this example, traffic arriving from interface service-ipsec1 exits through interface gigabitethernet0/0/0/3 and interface ATM 0/2/0/0.1. The destination of the traffic is at network 7.0.0.0/24.

If either one of the interfaces or the remote network goes down, the flow of traffic must stop. To do this, we use a Boolean AND expression.

```
track C1
type route reachability
 route ipv4 3.3.3.3/32
!
track C2
type route reachability
 route ipv4 1.2.3.4/32
track C3
type route reachability
 route ipv4 10.0.20.2/32
!
!
track C4
type route reachability
 route ipv4 10.0.20.0/24
!
track OBJ
type list boolean and
 object C1
 object C2
!
track OBJ2
type list boolean or
 object C1
 object C2
```

Configuring IPSLA based Object Tracking: Configuration Example

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

ACL configuration:

```
RP/0/RP0/CPU0:router(config) # ipv4 access-list abf-track
RP/0/RP0/CPU0:router(config-ipv4-acl) # 10 permit any nexthop track track1 1.2.3.4
```

Object tracking configuration:

```
RP/0/RP0/CPU0:router(config)# track track1
RP/0/RP0/CPU0:router(config-track)# type rtr 1 reachability
RP/0/RP0/CPU0:router(config-track)# delay up 5
```

```
RP/0/RP0/CPU0:router(config-track) # delay down 10
```

IPSLA configuration:

```
RP/0/RP0/CPU0:router(config) # ipsla
RP/0/RP0/CPU0:router(config-ipsla) # operation 1
RP/0/RP0/CPU0:router(config-ipsla-op) # type icmp echo
RP/0/RP0/CPU0:router(config-ipsla-icmp-echo) # source address 2.3.4.5
RP/0/RP0/CPU0:router(config-ipsla-icmp-echo) # destination address 1.2.3.4
RP/0/RP0/CPU0:router(config-ipsla-icmp-echo) # frequency 60
RP/0/RP0/CPU0:router(config-ipsla-icmp-echo) # exit
RP/0/RP0/CPU0:router(config-ipsla-op) # exit
RP/0/RP0/CPU0:router(config-ipsla) # schedule operation 1
RP/0/RP0/CPU0:router(config-ipsla-sched) # start-time now
RP/0/RP0/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 System Monitoring Configuration Guide for Cisco NCS 560 Series Routers
IP SLA commands	IP Service Level Agreement Commands on System Monitoring Command Reference for Cisco NCS 5500 Series Routers and Cisco NCS 540 and NCS 560 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: https://cfnng-stg.cisco.com/mibs.

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

Additional References



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.

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

- Prerequisites for Implementing Physical and Virtual Terminals, on page 53
- Information About Implementing Physical and Virtual Terminals, on page 53
- How to Implement Physical and Virtual Terminals on Cisco IOS XR Software, on page 56
- Configuration Examples for Implementing Physical and Virtual Terminals, on page 60

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*.

Line Templates

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

- Default line template—The default line template that applies to a physical and virtual terminal lines.
- Console line template—The line template that applies to the console line.

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

Line Template Configuration Mode

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

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

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

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

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

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



Note

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

Line Template Guidelines

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

• Modify the templates for the physical terminal lines on the router (the console port) from line template configuration mode. Use the **line console** command from XR Config mode to enter line template configuration mode for the console template.

• Modify the template for virtual lines by configuring a user-defined template with the **line** *template-name* command, configuring the terminal attributes for the user-defined template from line template configuration, and applying the template to a range of virtual terminal lines using the **vty pool** command.



Note

Before creating or modifying the vty pools, enable the telnet server using the **telnet server** command in XR Config mode. See Cisco IOS XR IP Addresses and Services Configuration Guide and Cisco IOS XR IP Addresses and Services Command Reference 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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	line {console default}	Enters line template configuration mode for the
	Example:	specified line template.
	<pre>RP/0/RP0/CPU0:router(config)# line console</pre>	• console —Enters line template configuration mode for the console template.
	or	default —Enters line template configuration mode for the default line
	<pre>RP/0/RP0/CPU0:router(config)# line default</pre>	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:	Saves configuration changes.
	• end	• When you issue the end command, the
	• commit	system prompts you to commit changes:
	Example:	
	•	Uncommitted changes found, commit them
	RP/0/RP0/CPU0:router(config-line)# end	before exiting(yes/no/cancel)?
	or	[cancel]:
	RP/0/RP0/CPU0:router(config-line)# 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.

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.

Creating and Modifying vty Pools

This task explains how to create and modify vty pools.

You can omit Step 3 to Step 5 (**line template** and **exit** commands) if you are configuring the default line template to reference a vty pool.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	telnet {ipv4 ipv6} server max-servers limit	Specifies the number of allowable Telnet
	Example:	servers. Up to 100 Telnet servers are allowed.
	RP/0/RP0/CPU0:router(config)# telnet ipv4 server max-servers 10	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:	Enters line template configuration mode for a user-defined template.
	<pre>RP/0/RP0/CPU0:router(config)# line template 1</pre>	
Step 4	Configure the terminal attribute settings for the specified line template using the commands in line template configuration mode.	
Step 5	exit	Exits line template configuration mode and
	Example:	returns the router to global configuration mode.
	RP/0/RP0/CPU0:router(config-line)# exit	

	Command or Action	Purpose
Step 6	vty-pool {default pool-name eem} first-vty	Creates or modifies vty pools.
	<pre>last-vty [line-template {default template-name}] Example:</pre>	• If you do not specify a line template with the line-template keyword, a vty pool defaults to the default line template.
	RP/0/RP0/CPU0:router(config)#vty-pool	• default —Configures the default vty poo
	default 0 5 line-template default Or	• The default vty pool must start at vt 0 and must contain a minimum of fiv vtys (vtys 0 through 4).
	RP/0/RP0/CPU0:router(config) #vty-pool pool1 5 50 line-template template1 OT	 You can resize the default vty pool by increasing the range of vtys that compose the default vty pool.
	RP/0/RP0/CPU0:router(config)#vty-pool eem 100 105 line-template template1	• pool-name —Creates a user-defined vty pool.
	RP/0/RP0/CPU0:router(config)#vty-pool default 0 5 line-template template1	 A user-defined pool must start at lea at vty 5, depending on whether the default vty pool has been resized.
		• If the range of vtys for the default vt pool has been resized, use the first range value free from the default lir 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 star with vty 10.
		• eem —Configures the embedded event manager pool.
		The default embedded event manage vty pool must start at vty 100 and must contain a minimum of six vty (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 are remains within the configuration session.
		end —Prompts user to take one of these action
		• Yes — Saves configuration changes and exits the configuration session.

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

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.

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

	Command or Action	Purpose
Step 3	(Optional) show users	Displays information about the active lines on the router.
Example:	Example:	
	RP/0/RP0/CPU0:router# show users	

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:

```
RP/0/RP0/CPU0:router# show running-config line console
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:

```
Line "con0_RP1_CPU0", Location "0/RP1/CPU0", Type "Console"
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 9600, "No" Parity, 2 stopbits, 8 databits
Template: console
Capabilities: Timestamp Enabled
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 100 105 line-template test3
```



Configuring Simple Network Management Protocol

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

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

- Prerequisites for Implementing SNMP, on page 63
- Restrictions for SNMP use on Cisco IOS XR Software, on page 63
- Information about Implementing SNMP, on page 64
- Session MIB support on subscriber sessions, on page 69
- How to Implement SNMP on Cisco IOS XR Software, on page 71

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 2³². 2³² is equal to 4.29 Gigabits.



Note

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

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

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

Information about Implementing SNMP

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

SNMP Functional Overview

The SNMP framework consists of three parts:

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

SNMP Manager

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

SNMP Agent

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

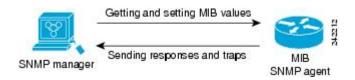
MIB

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

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

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

Figure 1: Communication Between an SNMP Agent and Manager



IP-MIB Support

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

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.

SNMP Versions

Cisco IOS XR software supports the following versions of SNMP:

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

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

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

SNMPv3 is a security model. A *security model* is an authentication strategy that is set up for a user and the group in which the user resides. A *security level* is the permitted level of security within a security model. A combination of a security model and a security level will determine which security mechanism is employed

when an SNMP packet is handled. See Security Models and Levels for SNMPv1, v2, v3, on page 66 for a list of security levels available in SNMPv3. The SNMPv3 feature supports RFCs 3411 to 3418.

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

Comparison of SNMPv1, v2c, and v3

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

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

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

Table 1: SNMPv1, v2c, and v3 Feature Support

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

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	HMAC-MD5 or HMAC-SHA	No	Provides authentication based on the HMAC ¹ -MD5 ² algorithm or the HMAC-SHA ³ .
v3	authPriv	HMAC-MD5 or HMAC-SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES ⁴ 56-bit encryption in addition to authentication based on the CBC ⁵ DES (DES-56) standard.
v3	authPriv	HMAC-MD5 or HMAC-SHA	3DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides 168-bit 3DES ⁶ level of encryption.
v3	authPriv	HMAC-MD5 or HMAC-SHA	AES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides 128-bit AES ⁷ level of encryption.

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

Use of 3DES and AES encryption standards requires that the security package (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 3: Order of Response Times from Least to Greatest

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

User-Based Security Model

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

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

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

USM uses two authentication protocols:

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

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

View-Based Access Control Model

The View-Based Access Control Model (VACM) enables SNMP users to control access to SNMP managed objects by supplying read, write, or notify access to SNMP objects. It prevents access to objects restricted by

views. These access policies can be set when user groups are configured with the **snmp-server group** command.

MIB Views

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

Access Policy

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

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

IP Precedence and DSCP Support for SNMP

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

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

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

Session MIB support on subscriber sessions

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

SNMP Notifications

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



Note

Inform requests (inform operations) are supported in Cisco IOS XR software.

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

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

Figure 2: Trap Received by the SNMP Manager

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

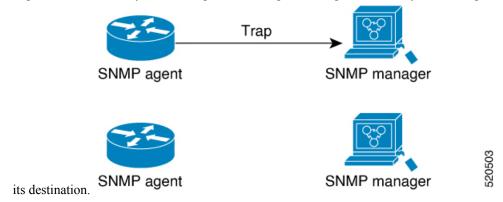
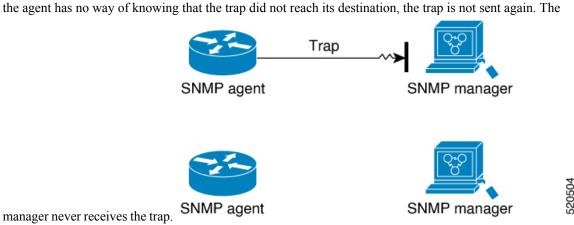


Figure 3: Trap Not Received by the SNMP Manager

In this illustration, the agent sends a trap to the manager, but the trap does not reach the manager. Because the agent has no way of knowing that the trap did not reach its destination, the trap is not sent again. The



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*.

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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	(Optional) snmp-server engineid local engine-id	Specifies the identification number of the local SNMP engine.
	Example:	
	RP/0/RP0/CPU0:router(config)# snmp-server engineID local 00:00:00:09:00:00:00:a1:61:6c:20:61	
Step 3	snmp-server view view-name oid-tree {included excluded}	Creates or modifies a view record.
	Example:	

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config)# snmp-server view view_name 1.3.6.1.2.1.1.5 included	
Step 4	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.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config) # snmp-server group group_name v3 noauth read view_name1 write view_name2</pre>	
Step 5	snmp-server user username groupname {v1 v2c v3 [auth {md5 sha} {clear encrypted} auth-password [priv des56 {clear encrypted} priv-password]]} [access-list-name]	Configures a new user to an SNMP group.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config) # snmp-server user noauthuser group_name v3</pre>	
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	Displays information about the status of
	Example:	SNMP.
	RP/0/RP0/CPU0:router# show snmp	
Step 8	(Optional) show snmp engineid	Displays information about the local SNMP
	Example:	engine.
	RP/0/RP0/CPU0:router# show snmp engineid	

	Command or Action	Purpose
Step 9	(Optional) show snmp group Example:	Displays information about each SNMP group on the network.
	RP/0/RP0/CPU0:router# show snmp group	
Step 10	(Optional) show snmp users Example:	Displays information about each SNMP username in the SNMP users table.
0, 44	RP/0/RP0/CPU0:router# show snmp users	
Step 11	(Optional) show snmp view Example:	Displays information about the configured views, including the associated MIB view family name, storage type, and status.
	RP/0/RP0/CPU0:router# show snmp view	

Configuring SNMPv3: Examples

Setting an Engine ID

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

```
config
snmp-server engineID local 00:00:00:00:00:00:00: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:

```
show snmp engineid
SNMP engineID 0000009000000alffffffff
```

Creating a View

There are two ways to create a view:

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

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

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

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

```
config
  snmp-server view SNMP VIEW1 1.3.6.1.2.1.1 included
```

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

```
config
  snmp-server view SNMP_VIEW1 1.3.6.1.2.1.1 included
  snmp-server view SNMP VIEW1 1.3.6.1.2.1.1.5 excluded
```

Verifying Configured Views

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

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

vldefault 1.3.6.1 - included nonVolatile active
SNMP_VIEW1 1.3.6.1.2.1.1 - included nonVolatile active
SNMP VIEW1 1.3.6.1.2.1.1.5 - excluded nonVolatile active
```

Creating Groups

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

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

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

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

Verifying Groups

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

Creating and Verifying Users

Given the following SNMPv3 view and SNMPv3 group configuration:

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

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

```
config
  snmp-server user noauthuser group name v3
```



Note

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

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

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

User name: noauthuser

Engine ID: 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_name v3 priv read view_name write view_name
```

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

RP/0/RP0/CPU0:router# snmp-server user authuser group name v3 auth md5 clear auth passwd



Note

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

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

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

User name: authuser

Engine ID: localSnmpID

storage-type: nonvolatile active
```

Given the following SNMPv3 view and SNMPv3 group configuration:

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

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

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



Note

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

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

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

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

Configuring SNMP Trap Notifications

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



Note

You can omit #unique_78 if you have already completed the steps documented under the #unique_78 task.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	snmp-server group name {v1 v2 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.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config) # snmp-server group group_name v3 noauth read view_name1 writer view_name2</pre>	
Step 3	snmp-server user username groupname {v1 v2c v3 {auth md5 sha} {clear encrypted} auth-password] [priv des56 {clear access-list-name]	Configures a new SNMP group or a table that maps SNMP users to SNMP views.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config) # snmp-server group group_name v3 noauth read view_name1 writer view_name2</pre>	
Step 4	[snmp-server host address [traps] [version {1 2c 3 [auth priv]}] community-string [udp-port port] [notification-type]	Specifies SNMP trap notifications, the version of SNMP to use, the security level of the notifications, and the recipient (host) of the
	Example:	notifications.
	RP/0/RP0/CPU0:router(config) # snmp-server host 12.26.25.61 traps version 3 noauth userV3noauth	

	Command or Action	Purpose
Step 5	<pre>snmp-server traps [notification-type] Example: RP/0/RP0/CPU0:router(config) # snmp-serve traps bgp</pre>	Enables the sending of trap notifications and specifies the type of trap notifications to be sent. • If a trap is not specified with the notification-type argument, all supported trap notifications are enabled on the router. To display which trap notifications are available on your router, enter the snmp-server traps? command.
Step 6	Use the commit or end command.	 commit — Saves the configuration changes and remains within the configuration session. end — Prompts user to take one of these actions: Yes — Saves configuration changes and exits the configuration session. No — Exits the configuration session without committing the configuration changes. Cancel — Remains in the configuration session, without committing the configuration changes.
Step 7	(Optional) show snmp host Example: RP/0/RP0/CPU0:router# show snmp host	Displays information about the configured SNMP notification recipient (host), port number, and security model.

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
```

In the following example, the output of the **show snmp host** commaand 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

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

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

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

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

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

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



Note

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

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

	Command or Action	Purpose
Step 2	(Optional) snmp-server contact system-contact-string	Sets the system contact string.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config)# snmp-server contact Dial System Operator at beeper # 27345</pre>	
Step 3	(Optional) snmp-server location system-location	Sets the system location string.
	Example:	
	RP/0/RP0/CPU0:router(config)# snmp-server location Building 3/Room 214	
Step 4	(Optional) snmp-server chassis-id serial-number	Sets the system serial number.
	Example:	
	RP/0/RP0/CPU0:router(config)# snmp-server chassis-id 1234456	
Step 5	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session.
		end —Prompts user to take one of these actions:
		• Yes — Saves configuration changes and exits the configuration session.
		• No —Exits the configuration session without committing the configuration changes.
		Cancel —Remains in the configuration session, without committing the configuration changes.

Defining the Maximum SNMP Agent Packet Size

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



Note

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

Procedure

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	(Optional) snmp-server packetsize byte-count	Sets the maximum packet size.
	Example:	
	RP/0/RP0/CPU0:router(config)# snmp-server packetsize 1024	
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.

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

	Command or Action	Purpose
Step 2	(Optional) snmp-server trap-source type interface-path-id	Specifies a source interface for trap notifications.
	Example:	
	RP/0/RP0/CPU0:router(config)# snmp-server trap-source POS 0/0/1/0	
Step 3	(Optional) snmp-server queue-length length	Establishes the message queue length for each notification
	Example:	normeation.
	RP/0/RP0/CPU0:router(config)# snmp-server queue-length 20	
Step 4	(Optional) snmp-server trap-timeout seconds	Defines how often to resend notifications on
	Example:	the retransmission queue.
	<pre>RP/0/RP0/CPU0:router(config)# snmp-server trap-timeout 20</pre>	
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.

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

	Command or Action	Purpose
	RP/0/RP0/CPU0:router# configure	
Step 2	Use one of the following commands: • snmp-server ipv4 precedence value • snmp-server ipv4 dscp value	Configures an IP precedence or IP DSCP value for SNMP traffic.
	Example: RP/0/RP0/CPU0:router(config) # snmp-server dscp 24	
Step 3	Use the commit or end command.	commit —Saves the configuration changes and remains within the configuration session.
		end —Prompts user to take one of these actions:
		• Yes — Saves configuration changes and exits the configuration session.
		• No —Exits the configuration session without committing the configuration changes.
		• Cancel —Remains in the configuration session, without committing the configuration changes.

Setting an IP Precedence Value for SNMP Traffic: Example

The following example shows how to set the SNMP 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
```

Displaying SNMP Context Mapping

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

Procedure

	Command or Action	Purpose
Step 1	show snmp context-mapping	Displays the SNMP context mapping table.
	Example:	
	RP/0/RP0/CPU0:router# show snmp context-mapping	

Monitoring Packet Loss

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

Before you begin



Note

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

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

	Command or Action	Purpose
Step 1	snmp-server mibs eventmib packet-loss type interface-path-id falling lower-threshold interval sampling-interval rising upper-threshold	Generates SNMP EVENT-MIB traps for the interface when the packet loss exceeds the specified thresholds. Up to 100 interfaces can be monitored.
	Example: RP/0/RP0/CPU0:router(config) # snmp-server mibs eventmib packet-loss TenGigE0/0/0/0 falling 1 interval 5 rising 2	falling lower-threshold —Specifies the lower threshold. When packet loss between two intervals falls below this threshold and an mteTriggerRising trap was generated previously, a SNMP mteTriggerFalling trap is generated. This trap is not generated until the packet loss exceeds the upper threshold and then falls back below the lower threshold.

Command or Action	Purpose
	interval sampling-interval — Specifies how often packet loss statistics are polled. This is a value between 5 and 1440 minutes, in multiples of 5.
	rising upper-threshold —Specifies the upper threshold. When packet loss between two intervals increases above this threshold, an SNMP mteTriggreRising trap is generated. This trap is not generated until the packet loss drops below the lower threshold and then rises above the upper threshold.

Configuring MIB Data to be Persistent

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

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

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

	Command or Action	Purpose
Step 1	(Optional) snmp-server mibs cbqosmib persist	Enables persistent storage of the CISCO-CLASS-BASED-QOS-MIB data.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config) # snmp-server mibs cbqosmib persist</pre>	
Step 2	snmp-server ifindex persist	Enables ifIndex persistence globally on all
	Example:	Simple Network Management Protocol (SNMP) interfaces.
	<pre>RP/0/RP0/CPU0:router(config) # snmp-server ifindex persist</pre>	

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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	snmp-server interface subset subset-number regular-expression expression	Enters snmp-server interface mode for the interfaces identified by the regular expression.
	<pre>Example: RP/0/RP0/CPU0:router(config) # snmp-server interface subset 10 regular-expression "^Gig[a-zA-Z]+[0-9/]+\." RP/0/RP0/CPU0:router(config-snmp-if-subset) #</pre>	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 <i>expression</i> argument must be entered surrounded by double quotes.
		Refer to the <i>Understanding Regular Expressions, Special Characters, and Patterns</i> module in for more information regarding regular expressions.
Step 3	notification linkupdown disable	Disables linkUp and linkDown traps for all
	Example:	interfaces being configured. To enable previously disabled interfaces, use the no form
	RP/0/RP0/CPU0:router(config-snmp-if-subset)# notification linkupdown disable	of this command
Step 4	Use the commit or end command.	commit —Saves the configuration changes, and remains within the configuration session.
		end —Prompts user to take one of these actions:
		• Yes — Saves configuration changes and exits the configuration session.
		• No —Exits the configuration session without committing the configuration changes.

	Command or Action	Purpose
		Cancel —Remains in the configuration mode, without committing the configuration changes.
Step 5	(Optional) show snmp interface notification subset subset-number	Displays the linkUp and linkDown notification status for all interfaces identified by the subset priority.
	Example:	priority.
	RP/0/RP0/CPU0:router# show snmp interface notification subset 10	
Step 6	(Optional) show snmp interface notification	Displays the linkUp and linkDown notification
	regular-expression expression Example:	status for all interfaces identified by the regular expression.
	<pre>RP/0/RP0/CPU0:router# show snmp interface notification regular-expression "^Gig[a-zA-Z]+[0-9/]+\."</pre>	
Step 7	(Optional) show snmp interface notification <i>type interface-path-id</i>	Displays the linkUp and linkDown notification status for the specified interface.
	Example:	
	RP/0/RP0/CPU0:router# show snmp interface notification tengige	

Configuring LinkUp and LinkDown Traps for a Subset of Interfaces



Configuring Periodic MIB Data Collection and Transfer

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

- Prerequisites for Periodic MIB Data Collection and Transfer, on page 89
- Information About Periodic MIB Data Collection and Transfer, on page 89
- How to Configure Periodic MIB Data Collection and Transfer, on page 91
- Periodic MIB Data Collection and Transfer: Example, on page 97

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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	snmp-server mib bulkstat object-list list-name	Defines an SNMP bulk statistics object list and enters bulk statistics object list configuration mode.
	Example:	
	<pre>snmp-server mib bulkstat object-list ifMib</pre>	
Step 3	add {oid object-name}	Adds a MIB object to the bulk statistics object
	Example:	list. Repeat as desired until all objects to be monitored in this list are added.
	RP/0/RP0/CPU0:router(config-bulk-objects) # add 1.3.6.1.2.1.2.2.1.11 RP/0/RP0/CPU0:router(config-bulk-objects) # add ifAdminStatus RP/0/RP0/CPU0:router(config-bulk-objects) # add ifDescr	list have to be indexed by the same MIB index. However, the objects in the
		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.

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

Configuring 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.

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	snmp-server mib bulkstat schema schema-name	Names the bulk statistics schema and enters bulk statistics schema mode.
	Example:	
	<pre>RP/0/RP0/CPU0:router(config) # snmp-server mib bulkstat schema intE0 RP/0/RP0/CPU0:router(config-bulk-sc) #</pre>	
Step 3	<pre>object-list list-name Example: RP/0/RP0/CPU0:router(config-bulk-sc) # object-list ifMib</pre>	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	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.

	Command or Action	Purpose
	• instance repetition oid max repeat-number Example: RP/0/RP0/CPU0:router(config-bulk-sc) # instance wild oid 1	 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.
	Or RP/0/RP0/CPU0:router(config-bulk-sc)# instance exact interface TenGigE 0/1.25 Or RP/0/RP0/CPU0:router(config-bulk-sc)#	The instance repetition command indicates data collection to repeat for a certain number of instances of a MIB object.
	<pre>instance range start 1 end 2 Or RP/0/RP0/CPU0:router(config-bulk-sc)# instance repetition 1 max 4</pre>	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	<pre>poll-interval minutes Example: RP/0/RP0/CPU0:router(config-bulk-sc)# poll-interval 10</pre>	Sets how often data should be collected from the object instances specified in this schema, in minutes. The default is once every 5 minutes. The valid range is from 1 to 20000.
Step 6	Use the commit or end command.	 commit — Saves the configuration changes and remains within the configuration session. end — Prompts user to take one of these actions: Yes — Saves configuration changes and exits the configuration session. No — Exits the configuration session without committing the configuration changes. Cancel — Remains in the configuration session, without committing the configuration changes.

Configuring Bulk Statistics Transfer Options

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

Before you begin

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

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	snmp-server mib bulkstat transfer-id transfer-id Example:	Identifies the transfer configuration with a name (<i>transfer-id</i> argument) and enters bulk statistics transfer configuration mode.
	<pre>RP/0/RP0/CPU0:router(config)# snmp-server mib bulkstat transfer bulkstat1</pre>	
Step 3	<pre>buffer-size bytes Example: RP/0/RP0/CPU0:router(config-bulk-tr)# buffersize 3072</pre>	(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.
	Durieroize 3072	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}	(Optional) Specifies the format of the bulk statistics data file (VFile). The default is schemaASCII.
	<pre>Example: RP/0/RP0/CPU0:router(config-bulk-tr)# format schemaASCII</pre>	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	<pre>schema schema-name Example: RP/0/RP0/CPU0:router(config-bulk-tr) # schema TenGigE RP/0/RP0/CPU0:router(config-bulk-tr) # schema TenGigE/0-CAR RP/0/RP0/CPU0:router(config-bulk-tr) #</pre>	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).

	Command or Action	Purpose
-	schema TenGigE	
Step 6	<pre>transfer-interval minutes Example: RP/0/RP0/CPU0:router(config-bulk-tr) # transfer-interval 20</pre>	(Optional) Specifies how often the bulk statistics file are transferred, in minutes. The default value is once every 30 minutes. The transfer interval is the same as the collection interval.
Step 7	<pre>url primary url Example: RP/0/RP0/CPU0:router(config-bulk-tr)# url primary ftp://user:password@host/folder/bulkstat1</pre>	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	<pre>url secondary url Example: RP/0/RP0/CPU0:router(config-bulk-tr)# url secondary tftp://10.1.0.1/tftpboot/user/bulkstat1</pre>	(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	<pre>retry number Example: RP/0/RP0/CPU0:router(config-bulk-tr)# retry 1</pre>	(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 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	<pre>retain minutes Example: RP/0/RP0/CPU0:router(config-bulk-tr)# retain 60</pre>	(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.

	Command or Action	Purpose
		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:	Begins the bulk statistics data collection and transfer process for this configuration.
	RP/0/RP0/CPU0:router(config-bulk-tr)# enable	For successful execution of this action, at least one schema with non-zero number of objects must be configured.
		 Periodic collection and file transfer begins only if this command is configured. Conversely, the no enable command stops the collection process. A subsequent enable starts the operations again.
		• Each time the collection process is started using the enable command, data is collected into a new bulk statistics file. When the no enable command is used, the transfer process for any collected data immediately begins (in other words, the existing bulk statistics file is transferred to the specified management station).
Step 12	commit minutes	If the maximum buffer size for a bulk statistics
	<pre>Example: RP/0/RP0/CPU0:router(config-bulk-tr)# retain 60</pre>	file is reached before the transfer interval time expires, the transfer operation is still initiated, but any bulk statistics data received after the file was full, and before it was transferred, are deleted. To correct this behavior, you can decrease the polling frequency, or increase the size of the bulk statistics buffer.
		If retain 0 is configured, no retries are attempted. This is because the interval between retries is the retain value divided by the retry value. For example, if retain 10 and retry 2 are configured, retries are attempted once every 5 minutes. Therefore, if you configure the retry command, you should also configure an appropriate value for the retain command.

Periodic MIB Data Collection and Transfer: Example

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

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

This example shows sample bulk statistics file content:

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

Periodic MIB Data Collection and Transfer: Example



Configuring Cisco Discovery Protocol

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

- Prerequisites for Implementing CDP, on page 99
- Information About Implementing CDP, on page 99
- How to Implement CDP on Cisco IOS XR Software, on page 101
- Configuration Examples for Implementing CDP, on page 104

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.



Note

CDP is an optional package. You must check if this package is installed on your system by running the command **show install active summary**.

Information About Implementing CDP

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

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

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

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

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

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

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

Table 4: Type-Length-Value Definitions for CDPv2

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

How to Implement CDP on Cisco IOS XR Software

Enabling CDP

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

	Command or Action	Purpose
Step 1	configure	Enters mode.
	Example:	
	RP/0/RP0/CPU0:router# configure	
Step 2	cdp	Enables CDP globally.
	Example:	
	RP/0/RP0/CPU0:router(config)# cdp	
Step 3	interface type interface-path-id	Enters interface configuration mode.
	Example:	
	RP/0/RP0/CPU0:router(config)# int TenGigE 0/5/0/11	
Step 4	cdp	Enables CDP on an interface.
	Example:	
	RP/0/RP0/CPU0:router(config-if)# cdp	
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.

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.

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

	Command or Action	Purpose
		Note A lower timer setting causes CDP updates to be sent more frequently. • In this example, CDP update packets are configured to be sent at a frequency of once every 20 seconds.
Step 5	Use the commit or end command. Example: RP/0/RP0/CPU0:router(config) # commit OR RP/0/RP0/CPU0:router(config) # end	 commit — Saves the configuration changes and remains within the configuration session. end — Prompts user to take one of these actions: Yes — Saves configuration changes and exits the configuration session. No — Exits the configuration session without committing the configuration changes. Cancel — Remains in the configuration session, without committing the configuration changes.
Step 6	(Optional) show cdp Example: RP/0/RP0/CPU0:router# show cdp	Displays global CDP information. The output displays the CDP version running on the router, the hold time setting, and the timer setting.

Monitoring CDP

This task shows how to monitor CDP.



Note

The commands can be entered in any order.

	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.
	RP/0/RP0/CPU0:router# show cdp entry *	

	Command or Action	Purpose
Step 2	show cdp interface [type interface-path-id location node-id]	Displays information about the interfaces on which CDP is enabled.
	Example:	
	RP/0/RP0/CPU0:router# show cdp interface pos 0/0/0/1	
Step 3	show cdp neighbors [type interface-path-id location node-id] [detail]	Displays detailed information about neighboring devices discovered using CDP.
	Example:	
	RP/0/RP0/CPU0:router# show cdp neighbors	
Step 4	show cdp traffic [location node-id]	Displays information about the traffic gathered
	Example:	between devices using CDP.
	RP/0/RP0/CPU0:router# show cdp traffic	

Configuration Examples for Implementing CDP

Enabling CDP: Example

The following example shows how to configure CDP globally and then enable CDP on Ethernet interface TenGigE 0/5/0/11:

```
cdp
interface 0/5/0/11
cdp
```

Modifying Global CDP Settings: Example

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

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

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

```
RP/0/RP0/CPU0:router# show cdp
Global CDP information:
  Sending CDP packets every 20 seconds
```

Sending a holdtime value of 30 seconds Sending CDPv2 advertisements is not enabled **Configuration Examples for Implementing CDP**



What is Smart Licensing?

Smart Licensing is a cloud-based, flexible software licensing model that enables you to activate and manage Cisco software licenses across their organization. Smart Licensing solution allows you to easily track the status of your license and software usage trends. Cisco Smart Licensing establishes a pool of licenses or entitlements that can be used across the entire organization in a flexible and automated manner. Smart Licensing helps simplify four core functions:

- **Purchase**—Creates a Smart Account (and optionally, your Virtual Account). Licenses are added to your Smart Account and are immediately available for use.
- Install—Register your product with your Smart Account using an account-based Registration Token. Thereafter, the entire process is automatic. Product Activation Keys (PAKs) and license files are no longer needed.
- Management—Make changes to license consumption by updating your configuration; any license change is automatically reflected in your Smart Account. You can share licenses in your Virtual Account through the license pooling option. License pools (logical grouping of licenses) can reflect your organization structure. Smart Licensing solution also offers Cisco Smart Software Manager, a centralized portal that enables you to manage all your Cisco software licenses from one centralized website.
- Visibility and Asset Management—Cisco Smart Software Manager (CSSM) portal offers an integrated view of the licenses you own and have deployed. You can use this data to make better purchase decisions, based on your consumption.
- What is Flexible Consumption Model?, on page 108
- Software Innovation Access, on page 111
- How Does Smart Licensing Work?, on page 114
- What is Cisco Smart Software Manager?, on page 116
- YANG Data Models for Smart Licensing, on page 118
- Configuring Smart Licensing, on page 119
- Registering and Activating Your Router, on page 126
- Verifying the Smart Licensing Configuration, on page 131

What is Flexible Consumption Model?

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Support for Flexible Consumption Model on A900-IMA-8Z-L-CC	Release 7.9.1	Support for Flexible Consumption Model (FCM) is now extended to the following Interface Module: • A900-IMA-8Z-L-CC-ASR 900 8 X 10GE Interface Module, Lite, Conformal Coated
Support for Flexible Consumption Model on • N560-IMA8Q • N560-IMA2C-DD • A900-IMA8Z	Release 7.4.1	Support for Flexible Consumption Model (FCM) is now extended to the following interface modules supported on the Cisco routers: • N560-IMA8Q • N560-IMA2C-DD • A900-IMA8Z
Essential and Advantage smart licenses in a combined entitlement	Release 7.4.1	With this release, the Advanced licenses are now referred to as the Advantage licenses, without essential entitlement. Also, a new license model — Advantage with Essentials, has been introduced that contains both Essential and Advantage licenses as a combined entitlement in a single PID. This simplifies the license procurement and management effort by eliminating the need to procure separate PIDs for Essential and Advantage licenses.

The Flexible Consumption Model (FCM) provides the capability and flexibility to purchase software capacity as needed. FCM delivers the following:

- Pay-as-you-grow—Enables you to lower initial costs and add more capacity over time.
- Simplify operations—FCM delivers the carrier-class IOS-XR software feature set with two software suites, Essentials and Advantage, that simplifies license management.
- Utilize capital efficiently—License pooling enables an efficient way to share licenses across the network.



Note

FCM is enabled by default for Cisco IOS XR Release 7.1.2 and later.

The types of licenses in this model are:

- Essential Licenses are the base licenses that are required by every active port for its operation. An example of Essential License is ESS-AC-10G-RTU-1.
- Advantage (earlier known as Advanced licenses) without essential licenses, are the licenses that are required on top of Essential Licenses for ports that use advanced features like L3VPN. An example of an advantage license is ADV-AC-10G-RTU-1.
- Advantage with Essential combination licenses are the licenses that are packaged together as a combined entitlement. Example of an advantage license with essentials is ADN-AC-100G-RTU-1.
- Tracking licenses, for example .

To enable Flexible Consumption model licensing on routers running Cisco IOS XR Releases 7.1.1 and earlier:

```
Router(config)# license smart flexible-consumption enable
Router(config)# commit
```

To verify the Flexible Consumption Model configuration:

Device# show running-config license smart flexible-consumption enable

The following table provides information about FCM licenses for Cisco NCS 560 routers:

Table 6: Flexible Consumption Licensing Model Usage Pattern

Flexible Consumption Model Licenses	Consumption Pattern
Essential Licenses: • ESS-AC-100G-RTU-1	If the router has a modular chassis, the license consumption checks are performed on the chassis.
Advantage Licenses:	
• ADV-AC-100G-RTU-1	
Hardware Tracking Licenses that support the chassis:	Licensing consumption is checked on every chassis.
• N560-4-SYS-TRK	
• N560-4-SYS-E-TRK	
• N560-4-SYS-CC-TRK	
• N560-4-SYSE-CC-TRK	

Flexible Consumption Model Licenses	Consumption Pattern
Hardware Tracking Licenses that support components in the chassis. • N560-IMA2C-TRK	Licensing consumption is checked on every chassis. For example, the N560-IMA2C-TRK license is tracked on the chassis that has this line card.
• A900-IMA8Z-TRK	
• A900-IMA8CS1ZM-TRK	

Table 7: Advantage with Essential Licenses

License Product ID	License Entitlement	License Description
ADN-AC-100G-RTU-1	1 ESS RTU Tag + 1 ADV RTU Tag	Access Advantage with Essentials SW RTU v1.0 100G
ESS-ADN-AC-100G-RT	1 ADV RTU Tag	Access Essentials to Advantage Upgrade RTU per 100G
ESS-ADN-AC100G-SST	1 ADV SIA 3 Tag	Access Essentials to Advantage Upgrade SIA 100G 1-35 month term
ESS-ADN-AC100G-S3	1 ADV SIA 3 Tag	Access Essentials to Advantage Upgrade SIA 100G 3-5 yrs
ESS-ADN-AC100G-S5	1 ADV SIA 3 Tag	Access Essentials to Advantage Upgrade SIA 100G 5-10 yrs
ADN-AC-100G-SIA-3	1 ESS SIA 3 Tag + 1 ADV SIA 3 Tag	Access Advantage with Essentials SIA 100G 3-5 year term
ADN-AC-100G-SIA-5	1 ESS SIA 3 Tag + 1 ADV SIA 3 Tag	Access Advantage with Essentials SIA 100G 5-10 year term
ADN-AC-100G-SIA-ST	1 ESS SIA 3 Tag + 1 ADV SIA 3 Tag	Access Advantage with Essentials SIA 100G 1-35 month term

Software Innovation Access

Table 8: Feature History Table

	Release Information	Feature Description
Software Innovation Access (SIA) Entitlement	Release 7.3.1	SIA license grants you access to the latest software upgrades which contain new features, bug fixes, and security enhancements for devices on your network. Also, it enables the consumption of Advantage and Essential Right-to-Use (RTU) licenses on your device, and allows portability of these RTU licenses from one device to another.

Overview

Software Innovation Access (SIA) subscription, a type of FCM licensing, provides access to the latest software upgrades and features for your network. SIA licenses enable the consumption of Right-to-Use (RTU) licenses for your devices to access software innovation and avail support for your devices throughout the term of subscription.

The benefits of SIA subscription are:

- Access to software innovation: SIA subscription provides access to continuous software upgrades which contain latest features, security enhancements, and bug fixes for all your devices at a network level.
- **Pooling of licenses**: SIA subscription enables Right-to-Use (RTU) licenses to be shared across your FCM network from a common license pool through the virtual account.
- **Protects your investment**: SIA subscription enables the portability of perpetual RTU licenses purchased for your current device to a next-generation router when you expand or upgrade your network.

The initial term of a SIA subscription is for a term of three years. You can renew the subscription by contacting your Cisco account representative. An equal number of SIA licenses and corresponding RTU licenses are required to enjoy the benefits, and ensure that your network is in compliance. There are two types of SIA licenses available:

- : To utilize Advantage RTU licenses, you need Advantage SIA licenses.
- : Essential SIA licenses are required to utilise Essential RTU on your device.

If your device is in a state of SIA Out-of-Compliance (OOC) the benefits cease.

SIA Out-of-compliance (OOC) state

When your device is in a SIA Out-of-compliance state, support for major software version upgrades in your network devices are restricted. However, you can continue to perform minor updates, SMU installations, and RPM installations, and continue to utilize the RTU licenses without support for porting.

A device can get into a SIA Out-of-compliance (OOC) state in the following instances:

- The SIA License *EVAL* period of 90 days has expired.
- The number of SIA licenses consumed has exceeded the number of SIA licenses purchased. This can also occur when the RTU licenses consumed is higher than the number of SIA licenses purchased.
- The term of the SIA license has expired and you haven't renewed the subscription.
- The license authorization status is:
 - Not Authorized: The license authorization code installed doesn't contain sufficient counts for the request. This can occur when you attempt to use more licenses than the licenses available in your Virtual Account.
 - Authorization expired: The device hasn't been able to connect to CSSM for an extended period, due to which the authorization status couldn't be verified.



Note

The CSSM smart license hierarchy applies to the Right-to-Use (RTU) license only. Therefore, if there is an insufficient RTU 100G license, CSSM can convert the RTU 400G license into four RTU 100G licenses. This is not applicable for SIA license.

In order to bring your device to In-Compliance state, perform one of the following steps:

- Register your device with CSSM if the SIA license EVAL period has expired.
- If the SIA license has expired or the number of SIA licenses consumed is more than the number of SIA licenses purchased, contact your Cisco Account Representative to purchase or renew the required licenses.
- If the authorization code has insufficient counts for the request, generate the code with sufficient counts.
- If the authorization has expired, connect the device with CSSM.

When the device enters an OOC state, a grace period of 90 days (cumulative of all the previous occurences) begins. During this period, SIA license benefits can still be availed. The system attempts to renew the authorization period by connecting with the CSSM during the grace period, or even after the grace period has expired. If an attempt isn't successful, it remains in an OOC state. If the attempt is successful, a new authorization period begins and the device is In-Compliance.

Verification

To verify the device compliance status, use the **show license platform summary** command:

Examples

Status: In-Compliance

		====	====
FCM	Total A900-IMA8CS1Z-M LCs in the system	1	0
FCM	Total A900-IMA8Z LCs in the system	2	0
FCM	NCS560-4 Base Hardware Tracking PID	1	0

Status: Out-of-Compliance (Grace Period Remaining)

Router# show license platform summary

		Coi	unt
Feature/Area	Entitlement	Last	Next
FCM	Total A900-IMA8CS1Z-M LCs in the system	1	0
FCM	Total A900-IMA8Z LCs in the system	2	0

Status: Out-of-Compliance (Grace Period expired and upgrades are blocked)

Router# show license platform summary

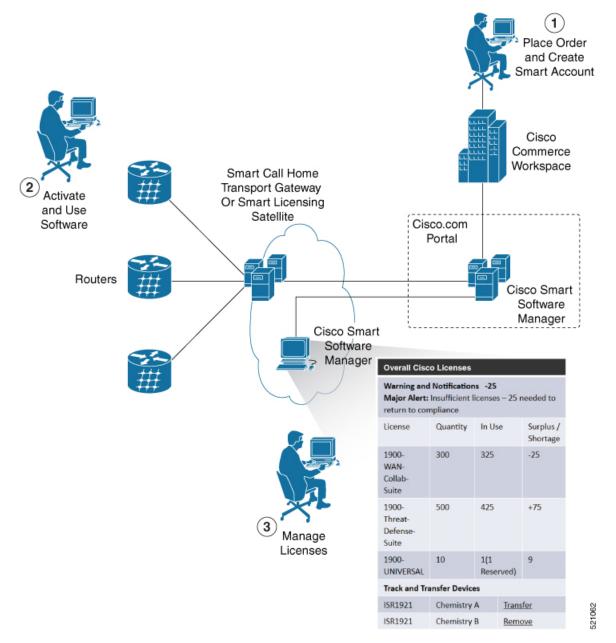
		Coi	ınt
Feature/Area	Entitlement	Last	Next
		====	
FCM	Total A900-IMA8CS1Z-M LCs in the system	1	0
FCM	Total A900-IMA8Z LCs in the system	2	0

How Does Smart Licensing Work?

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
Smart Transport Support	Release 7.4.1	You can now use Smart transport to communicate with CSSM. Smart transport is a transport method where a Smart Licensing (JSON) message is contained within an HTTPs message, and exchanged between a product instance and CSSM, to communicate. The following Smart transport configuration options are available: • Smart transport: In this method, a product instance uses a specific Smart transport licensing server URL. This must be configured exactly as shown in the workflow section. • Smart transport through an HTTPs proxy: In this method, a product instance uses a proxy server to communicate with the licensing server, and eventually, CSSM.

Figure 4: Smart Licensing - Workflow



- 1. Place Order and Create Smart Account—You must have a Smart Account to set up Smart Licensing.
 - **a.** Go to https://software.cisco.com/.
 - b. Under the Administration section, click Get a Smart Account or Request Access to an Existing Smart Account.
 - c. Verify or enter your Cisco.com profile details to complete creating a Smart Account.
- **2.** Activate and Use Software—Register your product. For more information, see the *Registering your Router* section. You can use either of the following options to communicate with the CSSM:

- Smart Transport—The Smart Transport method is a transport method where a Smart Licensing (JSON) message is contained within a HTTPs message, and exchanged between a product instance and CSSM to communicate. The following Smart transport configuration options are available.
 - Smart Transport: In this method, a product instance uses a specific Smart transport licensing server URL. This must be configured exactly as shown in the workflow section.
 - Smart transport through an HTTPs proxy: In this method, a product instance uses a proxy server to communicate with the licensing server, and eventually, CSSM.
- Smart Call Home—The Smart Call Home feature is automatically configured. Smart Call Home is used by Smart Licensing as a medium for communication with the CSSM. 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. For more information on Smart Call Home feature, see the Smart Call Home Deployment Guide.
- Smart Licensing CSSM On-Prem—The Smart licensing on-premise option provides an on-premises
 collector that can be used to consolidate and manage Smart license usage, as well as facilitate
 communications back to the CSSM at Cisco.com.



Note

Starting Cisco IOS XR Software Release 7.10.1, name server is required to validate CN in X.509 certificate is FQDN. TLS connection to CSSM is not be established if there is no name server configured. However, you can configure **crypto ca fqdn-check ip-address allow** to bypass the name-server configuration

You need to configure **crypto ca trustpoint Trustpool vrf** *vrf-name* with **http client vrf** *vrf-name* for CSSM communication in VRF.

3. Manage Licenses—You can manage and view reports about your overall license usage in the Smart Software Manager portal.

What is Cisco Smart Software Manager?

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). The Cisco Smart Software Manager allows you to:

- 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

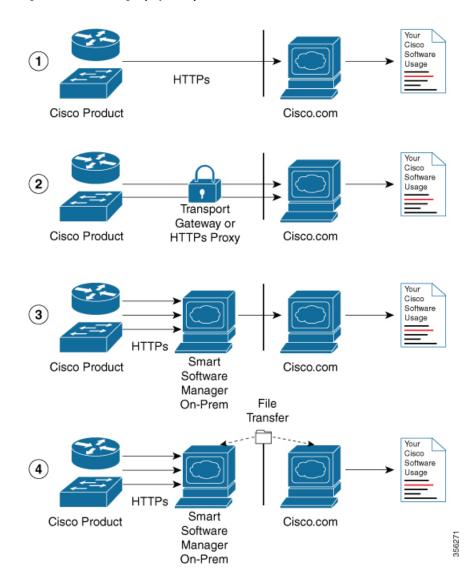
To access the Cisco Smart Software Manager:

- Go to https://software.cisco.com.
- Under the License section, click Smart Software Licensing.

Smart Licensing Deployment Options

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

Figure 5: Smart Licensing Deployment Options



1. Direct cloud access—In this method, Cisco products send usage information directly over the internet to CSSM on http://www.cisco.com; 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 CSSM 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 (once a month) to keep the databases in synchronization.

Options 1 and 2 provide easy deployment options, whereas options 3 and 4 provide secure environment deployment options.



Note

Smart Software On-Premise provides support for options 3 and 4.

The communication between Cisco devices and CSSM is facilitated by the Smart Call Home software.

YANG Data Models for Smart Licensing

YANG is a data modeling language that helps to create configurations, retrieve operational data and execute actions. The router acts on the data definition when these operations are requested using NETCONF RPCs. The data model handles the following types of requirements on the routers for smart licensing functionality:

Data	Data Model	CLI Commands
Configuration data: a set of writable data that is required to configure smart licensing on the router.	Native data model: Cisco-IOS-XR-smart-license-cfg.yang	 license smart reservation [no] license smart reservation license smart flexible-consumption enable [no] license smart flexible-consumption enable
Operational state data: a set of data that the system obtains at run time.	Common data model: cisco-smart-license.yang Native data model: Cisco-IOS-XR-smart- license-platform-oper.yang Cisco-IOS-XR-infrasmartlicense-oper.yang	show license platform summary show license platform detail show license [all summary usage udi]

Data	Data Model	CLI Commands
Actions: a set of NETCONF actions that support robust networkwide configuration transactions.	Native data model: Cisco-IOS-XR-smart-license-act.yang	 license smart register id token license smart deregister license smart renew id license smart renew auth license smart reservation request local license smart reservation cancel local license smart reservation install file <file path=""></file> license smart reservation return local license smart reservation return authorization file <file path=""></file> license smart transport smart license smart url <url> license smart software-upgrade enable license smart proxy hostname <hostname ip=""></hostname> license smart proxy port <port></port> </url>

You can access the data models from the Github repository. To learn more about the data models and put them to use, see the *Programmability Configuration Guide*.

Configuring Smart Licensing

Prerequisites for Configuring Smart Licensing

Ensure that you have completed the following activities on Cisco Smart Software Manager:

- Set up a Cisco Smart Account. For more information, see the *How Smart Licensing Works* section in this document.
- 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 10: 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:

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
	Example:	
	Router# configure terminal	
Step 2	call-home	Enters Call Home configuration mode.
	Example:	
	Router(config)# call-home	
Step 3	service active	Activates Call Home service.
	Example:	
	Router(config-call-home) # service active	

	Command or Action	Purpose
Step 4	<pre>contact-email-addr email-address Example: Router(config-call-home) # contact-email-addr username@example.com</pre>	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.
Step 5	<pre>profile CiscoTAC-1 Example: Router(config-call-home) # profile CiscoTAC-1</pre>	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. Note Starting Cisco IOS XR Software Release 7.10.1, name server is required to validate CN in X.509 certificate is FQDN. TLS connection to CSSM is not be established if there is no name server configured. However, you can configure crypto ca fqdn-check ip-address allow to bypass the name-server configuration You need to configure crypto ca trustpoint Trustpool vrf vrf-name with http client vrf vrf-name for CSSM communication in VRF.
Step 7	destination address http url Example: Router(config-call-home-profile) # destination address http https://tools.cisco.com/its/service/cobbe/services/DUEService	Connects the router to the Cisco Smart Software Manager.
Step 8	<pre>active Example: Router(config-call-home-profile) # active</pre>	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	<pre>commit Example: Router(config-call-home-profile) # commit</pre>	Commits the configuration.

	Command or Action	Purpose
Step 11	exit	Exits the Call Home destination profile
	Example:	configuration mode and returns to the Call Home configuration mode.
	Router(config-call-home-profile)# exit	
Step 12	exit	Exits the Call Home configuration mode and
	Example:	returns to the global configuration mode.
	Router(config-call-home) # exit Router(config) #	

Configuring a Connection Through an HTTP Proxy

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

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
	Example:	
	Router# configure terminal	
Step 2	call-home	Enters Call Home configuration mode.
	Example:	
	Router(config)# call-home	
Step 3	service active	Enables the Call Home feature.
	Example:	
	Router(config-call-home) # service active	
Step 4	contact-email-address email-address	Configures the default email address.
	Example:	
	Router(config-call-home)#	
	contact-email-addr sch-smart-licensing@cisco.com	
Step 5	http-proxy proxy-address port port-number	Provides the proxy server information to the
	Example:	Call Home service.
	Router(config-call-home)# http-proxy 198.51.100.10 port 3128	
Step 6	profile CiscoTAC-1	Enables the CiscoTAC-1 profile to be used
	Example:	with the Call Home service. By default, the
	Router(config-call-home)# profile CiscoTAC-1	CiscoTAC-1 profile is disabled.

Command or Action	Purpose
no destination transport-method email	Disables the email option for the Call Home
Example:	service.
Router(config-call-home-profile)# no destination transport-method email	
exit	Exits the Call Home destination profile
Example:	configuration mode and returns to the Call Home configuration mode.
<pre>Router(config-call-home-profile)# exit Router(config-call-home)#</pre>	The most service and the servi
profile profile-name	Enters the Call Home destination profile
Example:	configuration mode for the specified destination profile name. If the specified
Router(config-call-home) # profile test1	destination profile does not exist, it is created
reporting smart-licensing-data	Enables data sharing with the Call Home
Example:	service through the configured transport method, in this case, HTTP.
Router(config-call-home-profile)# reporting smart-licensing-data	mediod, in this edge, 111 11.
destination transport-method http	Enables the HTTP message transport method
Example:	
Router(config-call-home-profile)# destination transport-method http	
destination address http url	Connects the router to the Cisco Smart
Example:	Software Manager.
Router(config-call-home-profile)# destination address http https://tools.ciso.com/its/service/odite/services/DDEService	Note Starting Cisco IOS XR Software Release 7.10.1, name server is required to validate CN in X.509 certificate is FQDN. TLS connection to CSSM is not be established if there is no name server configured. However, you can configure crypto ca fqdn-check ip-address allow to bypass the name-server configuration You need to configure crypto ca trustpoint Trustpool vrf vrf-name with http client vrf vrf-name for CSSM communication in VRF.
active	Enables the destination profile.
Example:	
	no destination transport-method email Example: Router(config-call-home-profile) # no destination transport-method email exit Example: Router(config-call-home-profile) # exit Router(config-call-home) # profile profile-name Example: Router(config-call-home) # profile test1 reporting smart-licensing-data Example: Router(config-call-home-profile) # reporting smart-licensing-data destination transport-method http Example: Router(config-call-home-profile) # destination address http url Example: Router(config-call-home-profile) # destination address http https://tools.cisco.com/its/service/citie/services/IDEStervice/ active

	Command or Action	Purpose
Step 14	<pre>exit Example: Router(config-call-home-profile) # exit</pre>	Exits the Call Home destination profile configuration mode and returns to the Call Home configuration mode.
Step 15	<pre>exit Example: Router(config-call-home)# exit Router(config)#</pre>	Exits the Call Home configuration mode and returns to the global configuration mode.
Step 16	<pre>commit Example: Router(config) # commit</pre>	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.

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

	Command or Action	Purpose
	Router(config-call-home-profile) # destination address http http://209.165.201.15/haspartgate/as/services/leviceRequestiondler	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.
	Or Router(config-call-home-profile)# destination address http https://209.165.201.15/harportgatovay/services/haviceRequestionaler	Note Starting Cisco IOS XR Software Release 7.10.1, name server is required to validate CN in X.509 certificate is FQDN. TLS connection to CSSM is not be established if there is no name server configured. However, you can configure crypto ca fqdn-check ip-address allow to bypass the name-server configuration
		You need to configure crypto ca trustpoint Trustpool vrf vrf-name with http client vrf vrf-name for CSSM communication in VRF.
Step 7	no destination address http url	Removes the default destination address.
	Example:	
	Router(config-call-home-profile) # no destination address http https://tools.cisos.com/its/service/odde/services/DDEService	
Step 8	destination preferred-msg-format {long-text short-text xml}	(Optional) Configures a preferred message format. The default message format is XML.
	<pre>Example: Router(config-call-home-profile)# destination preferred-msg-format xml</pre>	
Step 9	active	Enables the destination profile.
	<pre>Example: Router(config-call-home-profile) # active</pre>	
Step 10	<pre>exit Example: Router(config-call-home-profile) # exit</pre>	Exits the Call Home destination profile configuration mode and returns to the Call Home configuration mode.
Step 11	<pre>exit Example: Router(config-call-home) # exit Router(config) #</pre>	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	Configures a source interface for the HTTP client.

	Command or Action	Purpose
	Example: Router(config) # http client source-interface ipv4 Vlan100	Note This command is mandatory for a VRF interface.
Step 13	crypto ca trustpool policy crl optional Example: Router(config) # crypto ca trustpool policy crl optional	(Optional) Bypasses the Certificate Revocation Lists (CRLs) check and establishes the connection. By default, the CRLs check is mandatory while establishing a TLS connection. We recommend this step when the smart licensing-enabled router is within a network and can rely on the License server to check the certificate status without retrieving and caching each CRL for every peer.
Step 14	<pre>commit Example: Router(config)# commit</pre>	Commits the configuration.
Step 15	<pre>end Example: Router(config) # end</pre>	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.



Note

When the router is in an unregistered state, the licenses are in EVAL (evaluation) mode. Evaluation period will last for 90 days.

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:

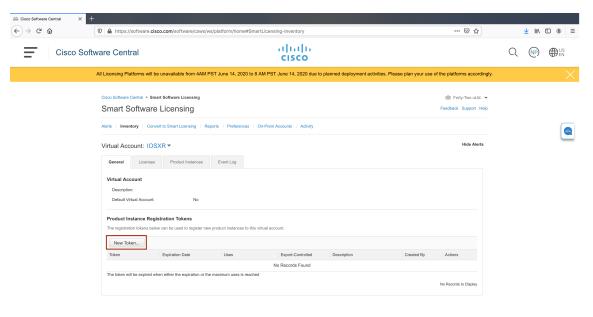
• Product specific: The same Registration Token can be used on different product types.

- · 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

Procedure

- 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://software.cisco.com/#.
- **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.



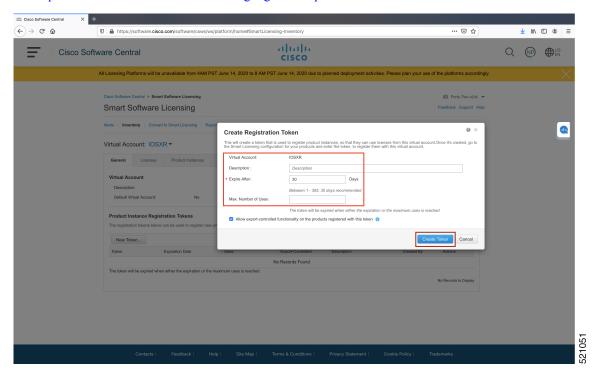
The **Create Registration Token** window is displayed.

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.

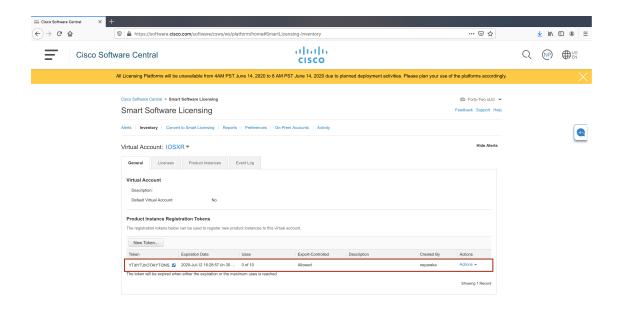
In the Max. Number of Uses field, enter the maximum number of uses allowed after which the token expires.

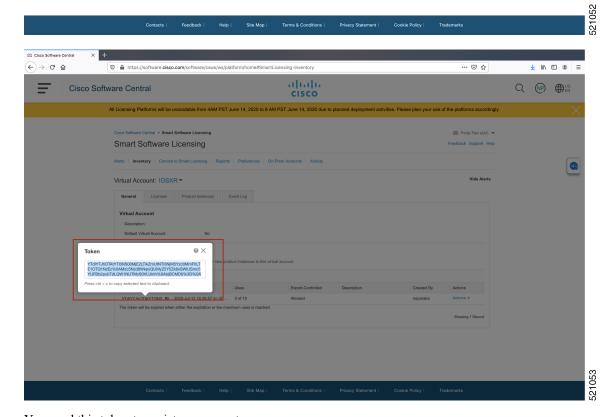
Select the **Allow export-controlled functionality on the products registered with this token** checkbox to ensure Cisco compliance with US and country-specific export policies and guidelines. For more information, see https://www.cisco.com/c/en/us/about/legal/global-export-trade.html.



Click Create Token.

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





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

Procedure

	Command or Action	Purpose
Step 1	license smart register idtoken token-ID Example: license smart register idtoken \$11404tnWebbsick\text{Webbs	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
	The second secon	license usage is displayed on the CSSM.

Example

Register Device Using YANG Data Model

As an alternative to the CLI command, the Cisco-IOS-XR-smart-license-act YANG data model can also be used to register your device with CSSM On-prem server. The following example shows the NETCONF RPC request:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <register-id-token xmlns="http://cisco.com/ns/yang/Cisco-IOS-XR-smart-license-act">
        <iid-token>$T14UytrNXBzbEs1ck8veUtWaG5abnZJOFdDa1FwbVRa%0AblRMbz0%3D%0A</id-token>
        </register-id-token>
    </rpc>
```

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.

Procedure

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

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.

Procedure

	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.

Verifying the Smart Licensing Configuration

Use the following **show** commands to verify the default Smart Licensing configuration. If any issue is detected, take corrective action before making further configurations.

	Command or Action	Purpose
Step 1	show license status	Displays the compliance status of Smart
	Example:	Licensing. Following are the possible status:
	Router# show license status	Waiting—Indicates that the initial state after your device has made a license entitlement request. The device establishes communication with Cisco and successfully registers itself with the Cisco license manager.

	Command or Action	Purpose
		• Authorized—Indicates that your device is able to communicate with the Cisco license manager, and is authorized to initiate requests for license entitlements.
		• Out-Of-Compliance—Indicates that one or more of your licenses are out-of-compliance. Buy more licenses, or renew the existing licenses.
		• Eval Period—Indicates that Smart Licensing is consuming the evaluation period. Register the device with the Cisco Licensing manager, else your license expires.
		Note Repetitive 'Smart Licensing evaluation expired' warning messages are displayed on the console every hour, but there is no functionality impact on the device. To stop these repetitive messages, register the device again with new a registration token.
		 Disabled—Indicates that Smart Licensing is disabled. Invalid—Indicates that Cisco does not recognize the entitlement tag as the tag is not in the database.
Step 2	show license all Example: Router# show license all	Displays all entitlements in use. The output also displays the associated licensing certificates, compliance status, Unique Device Identifier (UDI), and other details.
Step 3	show license status	Displays the status of all entitlements in use.
	Example: Router# show license status	
Step 4	show license udi Example: Router# show license udi	Displays the Unique Device Identifier (UDI) information.
Step 5	show license summary Example: Router# show license summary	Displays a summary of all entitlements in use.

	Command or Action	Purpose
Step 6	<pre>show license platform summary Example: Router# show license platform summary</pre>	Displays the registration status and provides detailed information about the essential, advantage without essentials, advantage with essentials, and tracking license consumption in generic or vortex license model.
Step 7	show license platform detail	Displays detailed information about:
	Example: Router# show license platform detail	 Licenses that can be consumed on a platform in both, generic and vortex models The active model, whether generic or vortex model The current count and the next consumption count of a license
Step 8	<pre>show call-home smart-licensing statistics Example: Router# show call-home smart-licensing statistics</pre>	Displays statistics of the communication between the Smart Licensing manager and the Cisco back-end using Smart Call Home. Note If the communication fails or drops, check your call home configuration for any errors.

Smart Licensing Configuration Examples

Example: Viewing the Call Home Profile

To display the **http Call Home profile** or the **On-Prem Call Home profile**, use the **show call-home profile** all command.

```
Router# show call-home profile all

Mon Jul 6 07:12:03.728 UTC

Profile Name: CiscoTAC-1
    Profile status: ACTIVE
    Profile mode: Full Reporting
    Reporting Data: Smart Call Home, Smart Licensing
    Preferred Message Format: xml
    Message Size Limit: 3145728 Bytes
    Transport Method: http
    HTTP address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService
    Other address(es): default

Periodic configuration info message is scheduled every 5 day of the month at 9:51
    Periodic inventory info message is scheduled every 5 day of the month at 9:36

Alert-group Severity
```

```
inventory
                              normal
   Syslog-Pattern
                          Severity
   ______
                             critical
Router# show call-home profile all
Tue Jul 7 18:48:47.325 UTC
Profile Name: CiscoTAC-1
   Profile status: ACTIVE
   Profile mode: Full Reporting
   Reporting Data: Smart Call Home, Smart Licensing
   Preferred Message Format: xml
   Message Size Limit: 3145728 Bytes
   Transport Method: http
   HTTP address(es): http://10.30.110.38/Transportgateway/services/RouterRequestHandler
   Other address(es): default
   Periodic configuration info message is scheduled every 5 day of the month at 9:51
   Periodic inventory info message is scheduled every 5 day of the month at 9:36
   Alert-group
                            Severity
   inventory
                             normal
   Syslog-Pattern
                           Severity
                                 critical
```

Example: Viewing License Information Before Registration

To display the license entitlements, use the show license all command:

```
Router# show license all
Mon Jul 6 07:25:24.369 UTC
Smart Licensing Status
Smart Licensing is ENABLED
Registration:
 Status: UNREGISTERED
 Export-Controlled Functionality: NOT ALLOWED
License Authorization:
 Status: EVAL EXPIRED on Jul 31 2019 10:11:28 UTC
Export Authorization Kev:
 Features Authorized:
   <none>
Utility:
 Status: DISABLED
Data Privacy:
 Sending Hostname: yes
   Callhome hostname privacy: DISABLED
   Smart Licensing hostname privacy: DISABLED
 Version privacy: DISABLED
```

```
Transport:
  Type: Callhome
License Usage
_____
(ESS-AC-100G-RTU-1):
 Description:
 Count: 1
 Version: 1.0
 Status: EVAL EXPIRED
 Export status: NOT RESTRICTED
(N560-4-SYS-E-TRK):
 Description:
  Count: 1
 Version: 1.0
 Status: EVAL EXPIRED
 Export status: NOT RESTRICTED
(A900-IMA8Z-TRK):
  Description:
 Count: 2
 Version: 1.0
 Status: EVAL EXPIRED
 Export status: NOT RESTRICTED
(A900-IMA8CS1ZM-TRK):
 Description:
 Count: 1
 Version: 1.0
 Status: EVAL EXPIRED
 Export status: NOT RESTRICTED
Product Information
_____
UDI: PID:NCS560-4, SN:FOX2305P81D
Agent Version
_____
Smart Agent for Licensing: 4.9.6 rel/41
Reservation Info
License reservation: DISABLED
To display the license usage information, use the show license usage command:
Router# show license usage
Mon Jul 6 07:26:07.579 UTC
License Authorization:
  Status: EVAL EXPIRED on Jul 31 2019 10:11:28 UTC
(ESS-AC-100G-RTU-1):
 Description:
  Count: 1
 Version: 1.0
 Status: EVAL EXPIRED
 Export status: NOT RESTRICTED
(N560-4-SYS-E-TRK):
 Description:
 Count: 1
 Version: 1.0
```

```
Status: EVAL EXPIRED
 Export status: NOT RESTRICTED
(A900-IMA8Z-TRK):
 Description:
  Count: 2
 Version: 1.0
 Status: EVAL EXPIRED
 Export status: NOT RESTRICTED
(A900-IMA8CS1ZM-TRK):
  Description:
  Count: 1
  Version: 1.0
 Status: EVAL EXPIRED
  Export status: NOT RESTRICTED
To display all the license summaries, use the show license summary command:
Router# show license summary
Mon Jul 6 07:27:23.696 UTC
Smart Licensing is ENABLED
Registration:
 Status: UNREGISTERED
 Export-Controlled Functionality: NOT ALLOWED
License Authorization:
  Status: EVAL EXPIRED
License Usage:
                        Entitlement tag
 License
                                                               Count Status
                           _____
                         (ESS-AC-100G-RTU-1)
                                                           1 EVAL EXPIRED
                         (N560-4-SYS-E-TRK)
                                                           1 EVAL EXPIRED
                         (A900-IMA8Z-TRK)
                                                            2 EVAL EXPIRED
                         (A900-IMA8CS1ZM-TRK)
                                                            1 EVAL EXPIRED
To display the license status information, use the show license status command:
Router# show license status
Mon Jul 6 07:28:23.414 UTC
Smart Licensing is ENABLED
Utility:
 Status: DISABLED
Data Privacy:
  Sending Hostname: yes
   Callhome hostname privacy: DISABLED
   Smart Licensing hostname privacy: DISABLED
 Version privacy: DISABLED
Transport:
  Type: Callhome
Registration:
 Status: UNREGISTERED
 Export-Controlled Functionality: NOT ALLOWED
License Authorization:
 Status: EVAL EXPIRED on Jul 31 2019 10:11:28 UTC
```

```
Export Authorization Key:
   Features Authorized:
     <none>
```

Example: Registering the Router

To register a device, use the license smart register idtoken command:

```
Router# license smart register idtoken T14UytrNXBzbEs1ck8veUtWaG5abnZJOFdDa1FwbVRa%0AblRMbz0%3D%0A
```

Example: Viewing License Information After Registration

To display the license entitlements, use the **show license all** command

```
Router# show license all
Mon Jul 6 07:13:23.823 UTC
Smart Licensing Status
_____
Smart Licensing is ENABLED
Registration:
 Status: REGISTERED
  Smart Account: Forty-Two uLtd.
 Virtual Account: IOSXR
 Export-Controlled Functionality: ALLOWED
  Initial Registration: SUCCEEDED on Jul 01 2020 07:14:24 UTC
  Last Renewal Attempt: None
  Next Renewal Attempt: Dec 28 2020 07:14:23 UTC
  Registration Expires: Jul 01 2021 07:01:42 UTC
License Authorization:
  Status: OUT OF COMPLIANCE on Jul 01 2020 07:14:38 UTC
  Last Communication Attempt: SUCCEEDED on Jul 05 2020 19:15:51 UTC
  Next Communication Attempt: Jul 06 2020 07:15:50 UTC
  Communication Deadline: Oct 03 2020 19:10:49 UTC
Export Authorization Key:
  Features Authorized:
    <none>
Utility:
  Status: DISABLED
Data Privacy:
  Sending Hostname: yes
   Callhome hostname privacy: DISABLED
   Smart Licensing hostname privacy: DISABLED
 Version privacy: DISABLED
Transport:
  Type: Callhome
License Usage
_____
Access Essentials SW Right-to-Use v1.0 per 100G (ESS-AC-100G-RTU-1):
  Description: Access Essentials SW Right-to-Use v1.0 per 100G
 Count: 1
 Version: 1.0
  Status: OUT OF COMPLIANCE
```

```
Export status: NOT RESTRICTED
Tracking PID for N560-4-SYS-E Vortex ATO (N560-4-SYS-E-TRK):
  Description: Tracking PID for N560-4-SYS-E Vortex ATO
  Count: 1
  Version: 1.0
 Status: AUTHORIZED
 Export status: NOT RESTRICTED
Tracking PID for ASR 900 8-port 10GE Interface Module (A900-IMA8Z-TRK):
  Description: Tracking PID for ASR 900 8-port 10GE Interface Module
  Count: 2
 Version: 1.0
 Status: AUTHORIZED
 Export status: NOT RESTRICTED
Tracking PID for ASR 900 8/16-port 1GE + 1-port 10GE Interface Module (A900-IMA8CS1ZM-TRK):
  Description: Tracking PID for ASR 900 8/16-port 1GE 1-port 10GE Interface
               Module
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
 Export status: NOT RESTRICTED
Product Information
UDI: PID:NCS560-4, SN:FOX2305P81D
Agent Version
_____
Smart Agent for Licensing: 4.9.6 rel/41
Reservation Info
License reservation: DISABLED
To display the license usage information, use the show license usage command:
Router# show license usage
Mon Jul 6 07:14:55.554 UTC
License Authorization:
 Status: OUT OF COMPLIANCE on Jul 01 2020 07:14:38 UTC
Access Essentials SW Right-to-Use v1.0 per 100G (ESS-AC-100G-RTU-1):
 Description: Access Essentials SW Right-to-Use v1.0 per 100G
  Count: 1
 Version: 1.0
  Status: OUT OF COMPLIANCE
 Export status: NOT RESTRICTED
Tracking PID for N560-4-SYS-E Vortex ATO (N560-4-SYS-E-TRK):
  Description: Tracking PID for N560-4-SYS-E Vortex ATO
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
 Export status: NOT RESTRICTED
Tracking PID for ASR 900 8-port 10GE Interface Module (A900-IMA8Z-TRK):
  Description: Tracking PID for ASR 900 8-port 10GE Interface Module
  Count: 2
 Version: 1.0
 Status: AUTHORIZED
 Export status: NOT RESTRICTED
```

```
Tracking PID for ASR 900 8/16-port 1GE + 1-port 10GE Interface Module (A900-IMA8CS1ZM-TRK):
  Description: Tracking PID for ASR 900 8/16-port 1GE 1-port 10GE Interface
              Module
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
  Export status: NOT RESTRICTED
To display all the license summaries, use the show license summary command:
Router# show license summary
Mon Jul 6 07:16:30.036 UTC
Smart Licensing is ENABLED
Registration:
 Status: REGISTERED
  Smart Account: Forty-Two uLtd.
 Virtual Account: IOSXR
 Export-Controlled Functionality: ALLOWED
  Last Renewal Attempt: None
 Next Renewal Attempt: Dec 28 2020 07:14:24 UTC
License Authorization:
  Status: OUT OF COMPLIANCE
  Last Communication Attempt: SUCCEEDED
  Next Communication Attempt: Jul 06 2020 19:15:57 UTC
License Usage:
 License
                        Entitlement tag
                                                       Count Status
  Access Essentials SW... (ESS-AC-100G-RTU-1)
                                                            1 OUT OF COMPLIANCE
                                                           1 AUTHORIZED
 Tracking PID for N56... (N560-4-SYS-E-TRK)
 Tracking PID for ASR... (A900-IMA8Z-TRK)
                                                            2 AUTHORIZED
 Tracking PID for ASR... (A900-IMA8CS1ZM-TRK)
                                                            1 AUTHORIZED
To display the license status information, use the show license status command:
Router# show license status
Mon Jul 6 07:17:27.729 UTC
Smart Licensing is ENABLED
Utility:
 Status: DISABLED
Data Privacy:
 Sending Hostname: yes
   Callhome hostname privacy: DISABLED
   Smart Licensing hostname privacy: DISABLED
 Version privacy: DISABLED
Transport:
 Type: Callhome
Registration:
 Status: REGISTERED
  Smart Account: Forty-Two uLtd.
  Virtual Account: IOSXR
  Export-Controlled Functionality: ALLOWED
  Initial Registration: SUCCEEDED on Jul 01 2020 07:14:24 UTC
 Last Renewal Attempt: None
 Next Renewal Attempt: Dec 28 2020 07:14:23 UTC
```

Registration Expires: Jul 01 2021 07:01:42 UTC

License Authorization:

Status: OUT OF COMPLIANCE on Jul 01 2020 07:14:38 UTC
Last Communication Attempt: SUCCEEDED on Jul 06 2020 07:15:57 UTC
Next Communication Attempt: Jul 06 2020 19:15:57 UTC
Communication Deadline: Oct 04 2020 07:10:56 UTC

Export Authorization Key:
 Features Authorized:
 <none>



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, on page 141
- Overview of FPD Image Upgrade Support, on page 141
- FPD upgrade service, on page 142

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.

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.



Note

Downgrade of FPDs is not recommended.



Note

FPD auto upgrade is not supported on this router.

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.
- Automatic FPD Image Upgrade (if enabled).
- 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.
Auto Upgrade	Upgrade using install SMU activation or during image upgrade. User can enable/disable auto upgrade feature.



Important

FPD auto upgrade is not supported on this router in releases before Cisco IOS XR Release 7.3.2.

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 0/0 0/0	NC55-18H18F NC55-18H18F NC55-18H18F	1.0 1.0 1.0	Bootloader IOFPGA SATA-M600-MCT		CURRENT CURRENT CURRENT	1.14 0.07 0.23	1.14 0.07 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.

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. Interface module (IMs) and RSPs 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. However, the **force** option must be used cautiously and only to recover a component from a failed upgrade.



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

- Migrate the software to a later Cisco IOS XR software release.
- Swap IMs or RSPs from a system running a different Cisco IOS XR software release.
- Insert a new IM or RSP.

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.

Configuration Examples for FPD Image Upgrade

The following examples indicates the use of commands associated with the FPD image upgrade procedure.

show fpd package Command Output: Example

Use the **show fpd package** command in System Admin EXEC mode to find out which IMs and RSPs are supported with your current Cisco IOS XR software release, which FPD image package you need for each IM or RSP, 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 a sample output from the **show fpd package** command:

A900-IMA8CS1Z-M	IMFPGA	YES	1.96	1.96	0.0
A900-IMA8Z	IMFPGA	YES	17.05	17.05	0.0
A900-IMA8Z-CC	IMFPGA	YES	17.05	17.05	0.0
A900-IMA8Z-L	IMFPGA	YES	1.45	1.45	0.0
A900-PWR1200-A	DCA-PriMCU(A) DCA-SecMCU(A)	NO NO	0.11	0.11	0.0
A900-PWR1200-D	LIT-PriMCU(A) LIT-SecMCU(A)	NO NO	2.04	0.04	0.0
A907-FAN-E	PSOC (A) PSOC (A)	NO NO	1.65 1.66	1.65 1.66	0.0
N560-4-FAN-H	PSOC (A)	NO	177.02	177.02	0.0
N560-4-FAN-H-CC	PSOC (A)	NO	177.02	177.02	0.0
N560-4-PWR-FAN	PSOC (A)	NO	177.08	177.08	0.0
N560-4-PWR-FAN-CC	PSOC (A)	NO	177.08	177.08	0.0
N560-4-RSP4	ADM(A) IOFPGA(A) PRIMARY-BIOS(A) SATA(A)	NO YES YES YES	1.06 0.64 0.17 2.10	1.06 0.64 0.17 2.10	0.0 0.0 0.0 0.0
N560-4-RSP4-CC	ADM(A) IOFPGA(A) PRIMARY-BIOS(A) SATA(A)	NO YES YES YES	1.06 0.64 0.17 2.10	1.06 0.64 0.17 2.10	0.0 0.0 0.0 0.0
N560-4-RSP4E	ADM(A) IOFPGA(A) PRIMARY-BIOS(A) SATA(A)	NO YES YES YES	1.06 0.64 0.17 2.10	1.06 0.64 0.17 2.10	0.0 0.0 0.0 0.0
N560-4-RSP4E-CC	ADM(A) IOFPGA(A) PRIMARY-BIOS(A) SATA(A)	NO YES YES YES	1.06 0.64 0.17 2.10	1.06 0.64 0.17 2.10	0.0 0.0 0.0 0.0
N560-FAN-H	PSOC (A)	NO	2.02	2.02	0.0
N560-IMA-8Q/4L	IMFPGA	YES	1.08	1.08	0.0
N560-IMA1W	CFP2-D-DCO CFP2-DE-DCO CFP2-DET-DCO CFP2-DETS-DCO CFP2-DS-DCO CFP2-DS100-DCO IMFPGA	NO NO NO NO NO NO YES	38.273 38.273 38.273 38.273 38.273 1.24		0.0 0.0 0.0
N560-IMA2C-CC	IMFPGA	YES	5.04	5.04	0.0
N560-PWR1200-D-E	QCS-PriMCU(A) QCS-SecMCU(A)	NO NO	1.82		0.0
N560-RSP4	ADM(A)	NO	1.06	1.06	0.0

	IOFPGA (A) PRIMARY-BIOS (A) SATA (A)	YES YES YES	0.64 0.17 2.10	0.64 0.17 2.10	0.0 0.0 0.0
N560-RSP4-E	ADM(A) IOFPGA(A) PRIMARY-BIOS(A) SATA(A)	NO YES YES YES	1.06 0.64 0.17 2.10	1.06 0.64 0.17 2.10	0.0 0.0 0.0 0.0
NCS560-IMA2C	IMFPGA	YES	5.04	5.04	0.0
NCS560-IMA2C-DD	IMFPGA	YES	1.24	1.24	0.0
NCS560-IMA2C-L	IMFPGA	YES	1.24	1.24	0.0

upgrade hw-module fpd Command Output: Example

Use the **upgrade hw-module fpd** command to upgrade the FPD image. 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/RP0/CPU0:ios# upgrade hw-module location 0/RP0 fpd ADM
Wed Oct 28 07:46:49.805 UTC
upgrade command issued (use "show hw-module fpd" to check upgrade status)
RP/0/RP0/CPU0:ios#RP/0/RP0/CPU0:Oct 28 07:46:51.949 UTC: optics driver[222]:
%PKT INFRA-FM-2-FAULT CRITICAL : ALARM CRITICAL :OPTICS RX POWER LANE-0 LOW ALARM :CLEAR :
Optics0/11/0/1: Optics0/11/0/1
0/RP0/ADMIN0:Oct 28 07:46:54.154 UTC: fpdserv[4899]: %INFRA-FPD Manager-1-UPGRADE ALERT :
Upgrade for the following FPDs has been committed:
O/RPO/ADMINO:Oct 28 07:46:54.154 UTC: fpdserv[4899]: %INFRA-FPD_Manager-1-UPGRADE_ALERT :
Location
                  FPD name
                                       Force
0/RP0/ADMIN0:Oct 28 07:46:54.154 UTC: fpdserv[4899]: %INFRA-FPD Manager-1-UPGRADE ALERT:
_____
0/RP0/ADMIN0:Oct 28 07:46:54.154 UTC: fpdserv[4899]: %INFRA-FPD Manager-1-UPGRADE ALERT :
                  ADM
                                       FALSE
0/RP0/ADMIN0:Oct 28 07:46:59.203 UTC: control driver[3690]: %INFRA-FPD Driver-6-UPGRADE RESULT
 : Upgrade completes 20 percent for fpd ADM@location 0/RP0.
RP/0/RP0/CPU0:ios#0/RP0/ADMIN0:Oct 28 07:47:09.204 UTC: control driver[3690]:
%INFRA-FPD Driver-6-UPGRADE RESULT : Upgrade completes 70 percent for fpd ADM@location
0/RP0/ADMIN0:Oct 28 07:47:10.854 UTC: control driver[3690]: %INFRA-FPD Driver-1-UPGRADE ALERT
 : FPD ADM@0/RP0 image programming completed with UPGRADE DONE state Info: [image 1.05 to
1.06 version)
0/RP0/ADMIN0:Oct 28 07:47:10.855 UTC: control driver[3690]: %INFRA-FPD Driver-1-UPGRADE ALERT
 : FPD ADM @location 0/RPO FPD upgraded and activated!
0/RP0/ADMIN0:Oct 28 07:47:10.857 UTC: shelf_mgr[3705]: %INFRA-SHELF_MGR-6-CARD SW OPERATIONAL
: Card: O/RPO software state going to Operational
0/RP0/ADMIN0:Oct 28 07:47:10.857 UTC: shelf mgr[3705]: %INFRA-SHELF MGR-6-CARD HW OPERATIONAL
: Card: 0/RPO hardware state going to Operational
RP/0/RP0/CPU0:Oct 28 07:47:17.931 UTC: fpd-serv[393]: %INFRA-FPD Manager-1-UPGRADE ALERT:
  FPD Upgrade Completed(use "show hw-module fpd" to check upgrade status)
O/RPO/ADMINO:Oct 28 07:47:19.155 UTC: fpdserv[4899]: %INFRA-FPD Manager-1-UPGRADE ALERT:
FPD Upgrade Completed (use "show hw-module fpd" to check upgrade status)
RP/0/RP0/CPU0:ios#
```

show platform Command Output: Example

Use the **show platform** command to verify that the IM is up and running.

```
RP/0/RP1/CPU0:router# show platform
Tue Oct 20 04:42:31.936 UTC
Node Type State Config state
```

0/0/CPU0	A900-IMA8CS1Z-M	OPERATIONAL	NSHUT	
0/1/CPU0	A900-IMA8CS1Z-M	OPERATIONAL	NSHUT	
0/2/CPU0	A900-IMA8CS1Z-M	OPERATIONAL	NSHUT	
0/4/CPU0	A900-IMA8Z-L	OPERATIONAL	NSHUT	
0/5/CPU0	A900-IMA8Z-L	OPERATIONAL	NSHUT	
0/7/CPU0	N560-IMA1W	OPERATIONAL	NSHUT	
0/9/CPU0	N560-IMA2C-DD	OPERATIONAL	NSHUT	
0/10/CPU0	A900-IMA8Z	OPERATIONAL	NSHUT	
0/11/CPU0	A900-IMA8Z-L	OPERATIONAL	NSHUT	
0/RP0/CPU0	N560-RSP4-E(Standby)	UNKNOWN	NSHUT	
0/RP1/CPU0	N560-RSP4-E(Active)	IOS XR RUN	NSHUT	
0/FT0/CPU0	N560-FAN-H	OPERATIONAL	NSHUT	
0/PM2/CPU0	A900-PWR1200-A	OPERATIONAL	NSHUT	

Auto FPD Upgrade

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
Auto FPD Upgrade	Release 7.3.2	This functionality enables automatic upgrade and reload for field-programmable devices (FPDs) whenever the Cisco IOS XR image has a newer FPD version. This functionality upgrades all route processors and line card FPDs simultaneously while displaying upgrade triggers on the console.

Effective Cisco IOS XR Release 7.3.2, you can enable automatic upgrade of FPD by using the "fpd auto-upgrade enable" command.



Note

Automatic upgrade of FPD is not supported in Cisco IOS XR Release 7.4.1.

To automatically upgrade all FPDs, use:

RP/0/RP0/CPU0:IOS(config) #fpd auto-upgrade enable

To reload the interface modules following the fpd auto-upgrade, use:

RP/0/RP0/CPU0:IOS(config) #fpd auto-reload enable

Limitations and Usage Guidelines

Limitations

- FPD auto-upgrade should be enabled only in the XR VM and *not* in the System Admin VM.
- With auto-upgrade enabled, if any card is in RELOAD REQUIRED state, auto-upgrade is re-triggered during any SSO or FPD-serv process restart.
- When an interface module (IM) or route processor (RP) is in RELOAD REQUIRED state and auto-upgrade is enabled, FPD upgrades are triggered again.

- With auto-upgrade enabled, if line card is inserted, an auto-upgrade is triggered. During this phase optics alarms are generated. If auto-reload is not enabled, you must reload the line cards manually to clear these alarms.
- SATA allows you to upgrade or downgrade when an FPD version change is available. Therefore, when auto-upgrade is enabled, the system automatically downgrades if lower versions are available. This behavior is specific only to SATA FPDs.
- FPD auto-reload is applicable for line cards only. Line cards are automatically reloaded after the fpd auto-upgrade process is completed.
- You must disable auto-upgrade during XR ISSU; otherwise, the router goes into a state where redundancy cannot be achieved. In this case, standby RP must be reloaded to achieve redundancy.
- TimingICs do not support **auto fpd upgrade** on NCS5500 Series Routers as the TimingIC requires a card reload immediately after upgrade. For the same reason, the TimingICs are not upgraded if the user specifies **location all** in the **auto fpd upgrade** command. To upgrade a TimingIC FPD, specify the FPD name along with the card location. For example, **upgrade hw-module fpd TimngIC-A location 0/RP0/cpu0.**

Usage Guidelines—Online Insertion of Line Cards

When a line card with a lower FPD version is inserted, one of the following scenarios apply:

- If fpd auto-upgrade and auto-reload are enabled, and a new line card is inserted, the system upgrades the line card FPDs automatically with the latest FPDs and reloads the line cards.
- If fpd auto-upgrade and auto-reload are both disabled, no action is required.
- If fpd auto-upgrade is enabled and auto-reload is disabled, the following alarms are displayed on the console:

```
RP/0/RP1/CPU0:Jun 1 10:05:46.095 UTC: optics_driver[231]: %PKT_INFRA-FM-3-FAULT_MAJOR: ALARM_MAJOR:OPTICS SUPPORTED_ERROR:DECLARE: Optics0/5/0/6: Optics0/5/0/6
RP/0/RP1/CPU0:Jun 1 10:05:46.096 UTC: optics_driver[231]: %PKT_INFRA-FM-2-FAULT_CRITICAL: ALARM_CRITICAL:OPTICS_NOT_SUPPORTED:DECLARE: Optics0/5/0/6: Optics0/5/0/6
```

You must reload the line cards manually to clear these alarms

Usage Guidelines—Online Insertion of RPs

When fpd auto-upgrade is enabled and a new RP is inserted, the system upgrades the RP FPDs automatically with the latest FPDs.



Note

RPs are not reloaded automatically. You must manually reload the RP or chassis for the latest FPD version to reflect.



Note

Reload of active RPs and line cards impacts the network traffic.

```
RP/0/RP0/CPU0:IOS# admin
Mon Jun 28 17:00:39.340 UTC
```

sysadmin-vm:0_RP1# hw-module location 0/RP1 reload
Mon Jun 28 17:00:52.178 UTC+00:00
Reload hardware module ? [no,yes] yes
#result Card graceful reload request on 0/RP1 succeeded.
RP/0/RP0/CPU0:IOS#

RP/0/RP0/CPU0:ios# show hw-module fpd
Fri Jun 4 10:08:01.784 UTC

Auto-upgrade: Enabled

					FPD Ve	ersions			
Location	Card type	HWver	FPD device	ATR Status	Running	Programd			
0/1	N560-IMA2C-DD	0.0	IMFPGA	CURRENT	1.27	1.27			
0/3	N560-IMA2C	0.0	IMFPGA	CURRENT	5.01	5.01			
0/5	A900-IMA8CS1Z-M	0.0	IMFPGA	CURRENT	1.98	1.98			
0/RP0	N560-4-RSP4	0.0	ADM	CURRENT	1.06	1.06			
0/RP0	N560-4-RSP4	0.0	IOFPGA	CURRENT	0.64	0.64			
0/RP0	N560-4-RSP4	0.0	PRIMARY-BIOS	CURRENT	0.18	0.18			
0/RP0	N560-4-RSP4	0.0	SATA	CURRENT	2.10	2.10			
0/RP1	N560-4-RSP4	0.0	ADM	CURRENT	1.06	1.06			
0/RP1	N560-4-RSP4	0.0	IOFPGA	CURRENT	0.64	0.64			
0/RP1	N560-4-RSP4	0.0	PRIMARY-BIOS	CURRENT	0.19	0.19			
0/RP1	N560-4-RSP4	0.0	SATA	CURRENT	2.10	2.10			
0/FT0	N560-4-PWR-FAN	0.1	PSOC	CURRENT	177.08	177.08			
0/FT1	N560-4-FAN-H	0.1	PSOC	CURRENT	177.02	177.02			
0/FT2	N560-4-FAN-H	0.1	PSOC	CURRENT	177.02	177.02			
RP/0/RP0/0	RP/0/RP0/CPU0:ios#								

Table 12: Action Required on FPDs After Auto Upgrade

FPD	Action Required
IOFPGA	Manual reload required
ADM	Upgraded version available immediately
PRIMARY-BIOS	Manual reload required
SATA	Upgraded version available immediately
PSOC	Upgraded version available immediately
IMFPGA	Manual reload required, if auto-reload is not configured

Configuring Auto FPD During System Upgrade

In case of Software upgrade (without ISSU), configure the **fpd auto-upgrade enable** command. All the FPDs are automatically upgraded in the currently installed image (V1). After the upgrade, the router automatically reloads and comes up with the new image (V2) with the upgraded FPDs already running. No additional reloads are required.



Note

System reloads are part of the SU process, therefore you can disable the FPD auto reload functionality by using the **fpd auto-reload disable** command.

1. Enable FPD auto-upgrade

RP/0/RP0/CPU0:IOS#conf

RP/0/RP0/CPU0:IOS(config) #fpd auto-upgrade enable

RP/0/RP0/CPU0:IOS#commit

2. Check for FPD Versions

 $RP/0/RP0/CPU0:Router\#show\ hw-module\ fpd$ Mon Jun 28 21:41:19.187 UTC

Auto-upgrade: Enabled

1 3						Versions
	Card type				Running	Programd
	NCS4200-1T16G-PS			CURRENT		
0/3	A900-IMA8CS1Z-M	0.0	IMFPGA	CURRENT	1.95	1.95
0/4	A900-IMA8Z	0.0	IMFPGA	CURRENT	17.05	17.05
0/5	A900-IMA8Z-L	0.0	IMFPGA	CURRENT	1.48	1.48
0/8	NCS4200-1T16G-PS	0.0	IMFPGA	CURRENT	1.98	1.98
0/9	N560-IMA1W	66.32	CFP2-DS-DCO	CURRENT	38.27397	38.27397
0/9	N560-IMA1W	0.0	IMFPGA	CURRENT	1.28	1.28
0/15	NCS4200-1T16G-PS	0.0	IMFPGA	CURRENT	1.98	1.98
0/RP0	N560-RSP4	0.0	ADM	CURRENT	1.06	1.06
0/RP0	N560-RSP4	0.0	IOFPGA	CURRENT	0.64	0.64
0/RP0	N560-RSP4	0.0	PRIMARY-BIOS	CURRENT	0.19	0.19
0/RP0	N560-RSP4	0.0	SATA	CURRENT	1.30	1.30
0/RP1	N560-RSP4	0.0	ADM	CURRENT	1.05	1.05
0/RP1	N560-RSP4	0.0	IOFPGA	CURRENT	0.64	0.64
0/RP1	N560-RSP4	0.0	PRIMARY-BIOS	CURRENT	0.19	0.19
0/RP1	N560-RSP4	0.0	SATA	CURRENT	1.30	1.30
0/FT0	N560-FAN-H	1.0	PSOC	CURRENT	2.02	2.02

3. Check that Auto Upgrades are Triggered for FPDs with Newer Versions Available



RP/0/RP0/CPU0:Router#

Note

At this step, all RSP, IMs, and fan FPD upgrades are initiated and completed. All cards are upgraded *before* the router reloads.

RP/0/RP1/CPU0:UUT-RSP4# copy tftp://<ncs560-mini-x-7.3.2.iso> harddisk:/RP/0/RP0/CPU0:IOS#install add source harddisk: ncs560-mini-x-7.3.2.iso ncs560-mcast-2.0.0.0-r732.x86_64.rpm ncs560-mgbl-2.0.0.0-r732.x86_64.rpm

```
ncs560-mpls-1.0.0.0-r732.x86 64.rpm
RP/0/RP0/CPU0:IOS#install ncs560-mini-x-7.3.2 ncs560-mcast-2.0.0.0-r732.x86_64
ncs560-mgbl-2.0.0.0-r732.x86 64 ncs560-mpls-1.0.0.0-r732.x86 64
RP/0/RP0/CPU0:IOS#install commit
RP/0/RP0/CPU0:ROUTER# install activate ncs560-mini-x-7.3.2.28I
Mon Jun 28 21:30:17.673 UTC
2021-06-28 21:30:20 Install operation 31 started by root123:
  install activate pkg ncs560-mini-x-7.3.2.28I
2021-06-28 21:30:20 Package list:
2021-06-28 21:30:20
                       ncs560-mini-x-7.3.2.28I
RP/0/RP0/CPU0:Jun 28 21:32:41.204 UTC: sdr instmgr[1213]: %PKT INFRA-FM-6-FAULT INFO:
INSTALL-IN-PROGRESS : DECLARE : 0/RPO/CPU0: INSTALL IN PROGRESS Alarm : being DECLARED for
the system
This install operation will reload the system, continue?
 [yes/no]:[yes] yes
2021-06-28 21:33:01 Install operation will continue in the background
RP/0/RP0/CPU0:ROUTER#RP/0/RP0/CPU0:Jun 28 21:41:40.910 UTC: fpd-serv[168]:
%PKT INFRA-FM-3-FAULT MAJOR : ALARM MAJOR :FPD-NEED-UPGRADE :DECLARE :0/RP0:
RP/0/RP0/CPU0:Jun 28 21:41:41.159 UTC: fpd-serv[168]: %PKT INFRA-FM-3-FAULT MAJOR:
ALARM MAJOR :FPD-NEED-UPGRADE :CLEAR :0/RP0:
0/RP0/ADMIN0:Jun 28 21:41:42.565 UTC: control driver[3205]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD SATA@0/RP0 image programming completed with
UPGRADE DONE state Info: [SDD firmware upgraded from 1.30 to 2.10]
0/RP0/ADMIN0:Jun 28 21:41:42.566 UTC: control driver[3205]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD SATA @location 0/RP0 FPD upgraded and activated!
0/RP0/ADMIN0:Jun 28 21:41:42.570 UTC: shelf mgr[3220]:
%INFRA-SHELF MGR-6-CARD SW OPERATIONAL : Card: 0/RPO software state going to Operational
0/RP0/ADMIN0:Jun 28 21:41:42.570 UTC: shelf mgr[3220]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/RPO hardware state going to Operational
RP/0/RP0/CPU0:Jun 28 21:41:42.486 UTC: fpd-serv[168]: %PKT_INFRA-FM-3-FAULT_MAJOR :
ALARM MAJOR : FPD-NEED-UPGRADE : DECLARE : 0/RP1:
0/RP1/ADMIN0:Jun 28 21:41:44.182 UTC: control driver[3220]:
%INFRA-FPD Driver-1-UPGRADE ALERT : FPD SATA@0/RP1 image programming completed with
UPGRADE DONE state Info: [SDD firmware upgraded from 1.30 to 2.10]
0/RP1/ADMIN0:Jun 28 21:41:44.182 UTC: control driver[3220]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD SATA @location 0/RP1 FPD upgraded and activated!
0/RP1/ADMIN0:Jun 28 21:41:48.905 UTC: control driver[3220]:
%INFRA-FPD Driver-6-UPGRADE RESULT: Upgrade completes 20 percent for fpd ADM@location
0/RP1.
0/RP1/ADMIN0:Jun 28 21:41:48.905 UTC: control driver[3220]:
%INFRA-FPD Driver-6-UPGRADE RESULT : Upgrade completes 50 percent for fpd
PRIMARY-BIOS@location 0/RP1.
0/RP1/ADMIN0:Jun 28 21:42:10.160 UTC: control driver[3220]:
%INFRA-FPD Driver-1-UPGRADE ALERT : FPD PRIMARY-BIOS@0/RP1 image programming completed
with UPGRADE DONE state Info: [ Upgrade Complete ]
0/RP1/ADMIN0:Jun 28 21:42:10.161 UTC: control driver[3220]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD PRIMARY-BIOS @location 0/RP1 upgrade completed.
0/RP1/ADMIN0:Jun 28 21:42:11.060 UTC: control driver[3220]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD ADM@0/RP1 image programming completed with UPGRADE
DONE state Info: [image 1.05 to 1.06 version]
0/RP1/ADMIN0:Jun 28 21:42:11.061 UTC: control driver[3220]:
%INFRA-FPD Driver-1-UPGRADE ALERT : FPD ADM @location 0/RP1 FPD upgraded and activated!
0/RP0/ADMIN0:Jun 28 21:42:11.062 UTC: shelf mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP1 software state going to Operational
0/RP0/ADMIN0:Jun 28 21:42:11.062 UTC: shelf mgr[3220]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/RP1 hardware state going to Operational
```

```
RP/0/RP0/CPU0:Jun 28 21:45:14.615 UTC: fpd_imfpga[121]: %INFRA-FPD_Driver-6-UPGRADE_RESULT
: Upgrade completes 50 percent for fpd IMFPGA@location 0/5.
RP/0/RP0/CPU0: Jun 28 21:45:14.616 UTC: fpd imfpga[121]: %INFRA-FPD Driver-6-UPGRADE RESULT
: Upgrade completes 50 percent for fpd IMFPGA@location 0/3.
RP/0/RP0/CPU0:Jun 28 21:48:24.763 UTC: fpd imfpga[121]: %INFRA-FPD Driver-6-UPGRADE RESULT
 : Upgrade completes 90 percent for fpd IMFPGA@location 0/5.
RP/0/RP0/CPU0:ROUTER#RP/0/RP0/CPU0:Jun 28 21:48:43.929 UTC: fpd imfpga[121]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD IMFPGA@0/5 image programming completed with
UPGRADE DONE state Info: [DONE ]
RP/0/RP0/CPU0:Jun 28 21:48:43.940 UTC: fpd imfpga[121]: %INFRA-FPD Driver-1-UPGRADE ALERT
: FPD IMFPGA @location 0/5 upgrade completed.
0/RP0/ADMIN0:Jun 28 21:48:45.354 UTC: shelf mgr[3220]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/5 hardware state going to Operational
0/RP0/ADMIN0:Jun 28 21:48:45.354 UTC: shelf mgr[3220]:
%INFRA-SHELF MGR-6-CARD SW OPERATIONAL : Card: 0/RP0 software state going to Operational
0/RP0/ADMIN0:Jun 28 21:48:45.354 UTC: shelf mgr[3220]:
%INFRA-SHELF_MGR-6-CARD_HW_OPERATIONAL : Card: 0/RPO hardware state going to Operational
P/0/RP0/CPU0:Jun 28 21:50:21.630 UTC: fpd imfpga[121]: %INFRA-FPD Driver-1-UPGRADE ALERT
 : FPD IMFPGA@0/3 image programming completed with UPGRADE DONE state Info: [DONE ]
RP/0/RP0/CPU0:Jun 28 21:50:21.642 UTC: fpd imfpga[121]: %INFRA-FPD Driver-1-UPGRADE ALERT
: FPD IMFPGA @location 0/3 upgrade completed.
0/RP0/ADMIN0:Jun 28 21:50:23.056 UTC: shelf mgr[3220]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/3 hardware state going to Operational
0/RP0/ADMIN0:Jun 28 21:50:23.056 UTC: shelf mgr[3220]:
%INFRA-SHELF MGR-6-CARD SW OPERATIONAL : Card: 0/RPO software state going to Operational
0/RP0/ADMIN0:Jun 28 21:50:23.057 UTC: shelf mgr[3220]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/RPO hardware state going to Operational
```

4. Check the RP FPD Versions and FPD Status

When the router is operational after the reload, all the RP, IMs, and fan FPDs are upgraded to the latest FPD versions.

 $\label{eq:sysadmin-vm:0_RP1\# show hw-module fpd} $$\operatorname{Mon Jun 28 21:51:20.187 UTC}$$

Auto-upgrade: Enabled

				FPD 7	Versions
Location	Card type	HWver FPD o	device ATR Status	Running	Programd
0/2	NCS4200-1T16G-PS	0.0 IMFP	GA CURRENT	1.98	1.98
0/3	A900-IMA8CS1Z-M	0.0 IMFP	GA CURRENT	1.98	1.98
0/4	A900-IMA8Z	0.0 IMFP	GA CURRENT	17.05	17.05
0/5	A900-IMA8Z-L	0.0 IMFP	GA CURRENT	1.49	1.49
0/8	NCS4200-1T16G-PS	0.0 IMFP	GA CURRENT	1.98	1.98
0/9	N560-IMA1W	66.32 CFP2-	DS-DCO CURRENT	38.27397	38.27397
0/9	N560-IMA1W	0.0 IMFP	GA CURRENT	1.28	1.28
0/15	NCS4200-1T16G-PS	0.0 IMFP	GA CURRENT	1.98	1.98

0/RP0	N560-RSP4	0.0	ADM	CURRENT	1.06	1.06
0/RP0	N560-RSP4	0.0	IOFPGA	CURRENT	0.64	0.64
0/RP0	N560-RSP4	0.0	PRIMARY-BIOS	CURRENT	0.19	0.19
0/RP0	N560-RSP4	0.0	SATA	CURRENT	2.10	2.10
0/RP1	N560-RSP4	0.0	ADM	CURRENT	1.05	1.05
0/RP1	N560-RSP4	0.0	IOFPGA	CURRENT	0.64	0.64
0/RP1	N560-RSP4	0.0	PRIMARY-BIOS	CURRENT	0.19	0.96
0/RP1	N560-RSP4	0.0	SATA	CURRENT	2.10	2.10
0/FT0	N560-FAN-H	1.0	PSOC	CURRENT	2.02	2.02

Configuring Auto FPD During ISSU

RP/0/RP0/CPU0:ROUTER#

ISSU occurs in two phases—in the System Admin VM and then in the XR VM.

In the System Admin VM mode, first execute the **fpd auto-upgrade enable** command. This configuration causes the FPDs on the route processor (RP) to automatically upgrade to the latest versions.

In the XR VM mode, you *must* disable the FPD auto upgrade. This is because IM FPGA upgrades are not automatically triggered as part of ISSU.



Note

Less than 50ms traffic loss is expected during ISSU.



Note

During both Admin and XR ISSU, always disable the auto reload functionality by using the **fpd auto-reload disable** command.

1. Enable FPD auto-upgrade in XR VM

RP/0/RP0/CPU0:IOS#conf
RP/0/RP0/CPU0:IOS(config)#fpd auto-upgrade enable
RP/0/RP0/CPU0:IOS#commit

2. Disable FPD auto-reload in XR VM

RP/0/RP0/CPU0:IOS#conf
RP/0/RP0/CPU0:IOS(config)#fpd auto-reload disable
RP/0/RP0/CPU0:IOS#commit

3. Check for FPD Versions

RP/0/RP0/CPU0:iso#show hw-module fpd Sun Aug 1 19:36:13.869 UTC Auto-upgrade:Enabled

0/11	NCS4200-8T-PS	0.0	IMFPGA	CURRENT	17.05	17.05
0/RP0	NCS4216-RSP-800	0.0	ADM	CURRENT	1.06	1.06
0/RP0	NCS4216-RSP-800	0.0	IOFPGA	CURRENT	0.78	0.78
0/RP0	NCS4216-RSP-800	0.0	PRIMARY-BIOS	CURRENT	0.21	0.21
0/RP0	NCS4216-RSP-800	0.0	SATA	CURRENT	2.20	2.20
0/RP1	NCS4216-RSP-800	0.0	ADM	CURRENT	1.06	1.06
0/RP1	NCS4216-RSP-800	0.0	IOFPGA	CURRENT	0.78	0.78
0/RP1	NCS4216-RSP-800	0.0	PRIMARY-BIOS	CURRENT	0.21	0.21
0/RP1	NCS4216-RSP-800	0.0	SATA	CURRENT	2.20	2.20
0/FT0	NCS4216-F2B-FAN	1.0	PSOC	CURRENT	2.02	2.02
0/PM0	A900-PWR1200-A	0.1	DCA-PriMCU	CURRENT	0.13	0.13
0/PM0	A900-PWR1200-A	0.1	DCA-SecMCU	CURRENT	2.03	2.03
0/PM1	A900-PWR1200-A	0.1	DCA-PriMCU	CURRENT	0.13	0.13
0/PM1 RP/0/RP0/	A900-PWR1200-A CPU0:iso#	0.1	DCA-SecMCU	CURRENT	2.03	2.03

4. Perform sysadmin ISSU Upgrade

Check that FPD upgrades are triggered as part of ISSU:

```
RP/0/RP0/CPU0:IOS#admin
sysadmin-vm:0 RP0# install extract ncs560-mini-x-7.3.2
Mon Aug 2 04:54:17.250 UTC+00:00
result Mon Aug 2 04:54:18 2021 Install operation 1 (install extract) started by user
 'cafyauto' will continue asynchronously.
sysadmin-vm:0 RPO# 0/RPO/ADMINO:Aug 2 04:58:47.990 UTC: inst mgr[4350]:
%INFRA-INSTMGR-2-OPERATION_SUCCESS : Install operation 1 completed successfully
Mon Aug 2 04:58:47 2021 Install operation 1 completed successfully.
sysadmin-vm:0 RP0# install activate issu host-7.3.2 ncs560-sysadmin-7.3.2
Mon Aug 2 05:26:31.442 UTC+00:00
op-initiated true
operation-id 2
result start-success
sysadmin-vm:0 RPO# 2021-08-02 05:26:31 Admin install operation 2 started
sysadmin-vm:0 RPO# 2021-08-02 05:26:31 install activate issu host-7.3.2
ncs560-sysadmin-7.3.2
sysadmin-vm:0 RPO# 2021-08-02 05:26:35 Install operation 2 started: ISSU prepare
install activate issu host-7.3.2 ncs560-sysadmin-7.3.2
sysadmin-vm:0 RP0# 0/RP0/ADMIN0:Aug 2 05:30:22.441 UTC: inst mgr[4350]:
%INFRA-INSTMGR-2-OPERATION SUCCESS: Install operation 2 completed successfully
2021-08-02 05:30:22 Install operation 2 completed successfully
sysadmin-vm:0 RPO# 2021-08-02 05:30:23 Admin install operation 3 started
sysadmin-vm:0 RPO# 2021-08-02 05:30:23 install activate issu host-7.3.2
ncs560-sysadmin-7.3.2
sysadmin-vm:0 RPO# 2021-08-02 05:30:25 Install operation 3 started: ISSU activate
install activate issu host-7.3.2 ncs560-sysadmin-7.3.2
sysadmin-vm:0 RPO#
sysadmin-vm:0 RP0# 2021-08-02 05:31:10 Admin VM of node 0/RP1 will now reload as part
of ISSU operation 3
```

```
O/RPO/ADMINO:Aug 2 05:31:17.314 UTC: shelf mgr[3212]: %INFRA-SHELF MGR-4-VM RELOAD:
Reloading VM on 0/RP1
0/RP0/ADMIN0:Aug 2 05:32:12.842 UTC: shelf mgr[3212]:
%INFRA-SHELF MGR-6-CARD SW OPERATIONAL : Card: 0/RP1 software state going to Operational
RP/0/RP0/CPU0:Aug 2 05:33:05.763 UTC: fpd-serv[419]: %PKT INFRA-FM-3-FAULT MAJOR:
ALARM MAJOR : FPD-NEED-UPGRADE : DECLARE : 0/RP1:
sysadmin-vm:0 RPO# 2021-08-02 05:33:28 Admin VM of node 0/RPO will now reload as part
of ISSU operation 3
RP/0/RP0/CPU0:Aug 2 05:35:44.331 UTC: fpd-serv[419]: %PKT INFRA-FM-3-FAULT MAJOR:
ALARM MAJOR : FPD-NEED-UPGRADE : DECLARE : 0/RP0:
O/RP1/ADMINO:Aug 2 05:36:01.213 UTC: inst mgr[6351]: %INFRA-INSTMGR-2-OPERATION SUCCESS
 : Install operation 3 completed successfully
0/RP0/ADMIN0:Aug 2 05:36:29.242 UTC: control driver[3920]:
%INFRA-FPD Driver-1-UPGRADE ALERT : Auto fpd triggered for FPD SATA upgrade @ location
0/RPO please wait until all fpd upgrades are done
0/RP0/ADMIN0:Aug 2 05:36:29.545 UTC: control driver[3920]:
%INFRA-FPD Driver-1-UPGRADE ALERT : FPD SATA@0/RP0 image programming completed with
UPGRADE DONE state Info: [SDD firmware upgraded from 1.30 to 2.10]
0/RP0/ADMIN0:Aug 2 05:36:29.545 UTC: control driver[3920]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD SATA @location 0/RPO FPD upgraded and activated!
0/RP1/ADMIN0:Aug 2 05:36:29.557 UTC: shelf mgr[4032]:
%INFRA-SHELF MGR-6-CARD SW OPERATIONAL : Card: 0/RPO software state going to Operational
0/RP1/ADMIN0:Aug 2 05:36:29.557 UTC: shelf mgr[4032]:
%INFRA-SHELF MGR-6-CARD_HW_OPERATIONAL : Card: 0/RP0 hardware state going to Operational
0/RP1/ADMIN0:Aug     2 05:36:29.667 UTC: control_driver[4015]:
%INFRA-FPD Driver-1-UPGRADE ALERT : Auto fpd triggered for FPD SATA upgrade @ location
0/RP1 please wait until all fpd upgrades are done
0/RP1/ADMIN0:Aug 2 05:36:29.981 UTC: control driver[4015]:
%INFRA-FPD Driver-1-UPGRADE ALERT : FPD SATA@0/RP1 image programming completed with
UPGRADE DONE state Info: [SDD firmware upgraded from 1.30 to 2.10]
0/RP1/ADMIN0:Aug 2 05:36:29.981 UTC: control driver[4015]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD SATA @location 0/RP1 FPD upgraded and activated!
0/RP1/ADMIN0:Aug 2 05:36:29.987 UTC: shelf mgr[4032]:
%INFRA-SHELF MGR-6-CARD SW OPERATIONAL : Card: 0/RP1 software state going to Operational
0/RP1/ADMIN0:Aug 2 05:36:29.987 UTC: shelf mgr[4032]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/RP1 hardware state going to Operational
0/RP1/ADMIN0:Aug  2 05:36:30.059 UTC: control_driver[4015]:
%INFRA-FPD Driver-1-UPGRADE ALERT : Auto fpd triggered for FPD PRIMARY-BIOS upgrade @
location 0/RP1 please wait until all fpd upgrades are done
0/RP0/ADMIN0:Aug 2 05:36:31.704 UTC: control driver[3920]:
%INFRA-FPD Driver-1-UPGRADE ALERT : Auto fpd triggered for FPD PRIMARY-BIOS upgrade @
location 0/RPO please wait until all fpd upgrades are done
0/RP1/ADMIN0:Aug 2 05:36:35.977 UTC: control driver[4015]:
%INFRA-FPD Driver-6-UPGRADE RESULT : Upgrade completes 50 percent for fpd
PRIMARY-BIOS@location 0/RP1.
0/RP0/ADMIN0:Aug 2 05:36:36.706 UTC: control driver[3920]:
%INFRA-FPD Driver-6-UPGRADE RESULT : Upgrade completes 50 percent for fpd
PRIMARY-BIOS@location 0/RP0.
0/RP1/ADMIN0:Aug 2 05:36:55.985 UTC: control driver[4015]:
%INFRA-FPD Driver-6-UPGRADE RESULT : Upgrade completes 100 percent for fpd
PRIMARY-BIOS@location 0/RP1.
0/RP1/ADMIN0:Aug 2 05:36:56.012 UTC: control driver[4015]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD PRIMARY-BIOS@0/RP1 image programming completed
with UPGRADE DONE state Info: [ Upgrade Complete ]
0/RP1/ADMIN0:Aug 2 05:36:56.012 UTC: control driver[4015]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD PRIMARY-BIOS @location 0/RP1 upgrade completed.
```

```
0/RP1/ADMIN0:Aug 2 05:36:56.018 UTC: shelf mgr[4032]:
%INFRA-SHELF MGR-6-CARD SW OPERATIONAL : Card: 0/RP1 software state going to Operational
0/RP1/ADMIN0:Aug 2 05:36:56.018 UTC: shelf mgr[4032]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/RP1 hardware state going to Operational
0/RP0/ADMIN0:Aug 2 05:36:56.582 UTC: control driver[3920]:
%INFRA-FPD Driver-1-UPGRADE ALERT : FPD PRIMARY-BIOS@0/RP0 image programming completed
with UPGRADE DONE state Info: [ Upgrade Complete ]
0/RP0/ADMIN0:Aug 2 05:36:56.869 UTC: control driver[3920]:
%INFRA-FPD Driver-1-UPGRADE ALERT: FPD PRIMARY-BIOS @location 0/RPO upgrade completed.
0/RP1/ADMIN0:Aug 2 05:36:56.871 UTC: shelf mgr[4032]:
%INFRA-SHELF MGR-6-CARD SW OPERATIONAL : Card: 0/RPO software state going to Operational
0/RP1/ADMIN0:Aug 2 05:36:56.872 UTC: shelf mgr[4032]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/RPO hardware state going to Operational
RP/0/RP0/CPU0:IOS#admin
Mon Aug 2 05:49:27.392 UTC
cafyauto connected from 192.0.0.4 using ssh on sysadmin-vm:0 RP1
sysadmin-vm:0 RP1# install commit
Mon Aug 2 05:49:40.694 UTC+00:00
result Mon Aug 2 05:49:41 2021 Install operation 4 (install commit) started by user
'cafyauto' will continue asynchronously.
sysadmin-vm:0 RP1# Mon Aug 2 05:49:44 2021 Install operation 4 completed successfully.
sysadmin-vm:0_RP1# 0/RP1/ADMIN0:Aug 2 05:49:44.733 UTC: inst_mgr[6351]:
%INFRA-INSTMGR-2-OPERATION SUCCESS: Install operation 4 completed successfully
0/RP1/ADMIN0:Aug 2 05:49:46.936 UTC: inst mgr[6351]: %PKT INFRA-FM-6-FAULT INFO :
INSTALL-IN-PROGRESS : CLEAR : 0/RP1: Calvados INSTALL IN PROGRESS Alarm : being CLEARED
for the system
sysadmin-vm:0 RP1#
```

5. Check the FPD Versions

6. Disable FPD auto-upgrade in XR VM

```
RP/0/RP1/CPU0:PP-Router#conf
RP/0/RP1/CPU0:PP-Router(config)#fpd auto-upgrade disable
RP/0/RP1/CPU0:PP-Router(config)#commit
```

7. Perform ISSU Upgrade in XR VM

```
RP/0/RP0/CPU0:IOS#install extract ncs560-mini-x-7.3.2
Mon Aug 2 05:55:19.880 UTC
2021-08-02 05:55:22 Install operation 100 started by cafyauto:
  install extract ncs560-mini-x-7.3.2
2021-08-02 05:55:22 Package list:
2021-08-02 05:55:22
                       ncs560-mini-x-7.3.2
2021-08-02 05:55:23 Install operation will continue in the background
RP/0/RP0/CPU0:IOS#2021-08-02 05:59:42 Install operation 100 finished successfully
RP/0/RP0/CPU0:Aug 2 05:59:42.840 UTC: sdr_instmgr[1213]:
%INSTALL-INSTMGR-2-OPERATION SUCCESS: Install operation 100 finished successfully
RP/0/RP0/CPU0:IOS#install activate issu ncs560-xr-7.3.2
Mon Aug 2 06:02:06.097 UTC
2021-08-02 06:02:08 Install operation 101 started by cafyauto:
  install activate issu ncs560-xr-7.3.2
2021-08-02 06:02:08 Package list:
2021-08-02 06:02:08
                       ncs560-xr-7.3.2
This install operation will start the issu, continue?
[yes/no]:[yes] yes
2021-08-02 06:04:03 Install operation will continue in the background
RP/0/RP0/CPU0:IOS#RP/0/RP0/CPU0:Aug 2 06:04:11.315 UTC: issudir[342]:
```

```
%PKT INFRA-FM-6-FAULT INFO : ISSU-IN-PROGRESS :DECLARE :0/RP0/CPU0: ISSU IN PROGRESS
Alarm : being DECLARED for the system
RP/0/RP0/CPU0:IOS# RP/0/RP1/CPU0:Aug 2 06:18:13.640 UTC: imfpga[345]: Change ISSU
Node State from [Secondary] to [Primary]
RP/0/RP1/CPU0: Aug 2 06:18:13.659 UTC: tmgctrl[444]: ISSU Node Role Change Notification
 [Primary]
RP/0/RP1/CPU0:Aug 2 06:18:13.725 UTC: issudir[139]:
%INSTALL-ISSU INFRA-1-OPERATION ISSU RUN DONE : ISSU Run has completed for install
operation '101'. Issue the command "install activate issu cleanup" to continue the
operation.
RP/0/RP1/CPU0:Aug 2 06:18:13.729 UTC: issudir[139]: %PKT INFRA-FM-6-FAULT INFO:
ISSU-IN-PROGRESS : DECLARE : 0/RP1/CPU0:
RP/0/RP1/CPU0:Aug 2 06:19:13.695 UTC: fpd-serv[216]: %PKT INFRA-FM-3-FAULT MAJOR :
ALARM MAJOR : FPD-NEED-UPGRADE : DECLARE : 0/3:
RP/0/RP1/CPU0:Aug 2 06:19:13.695 UTC: fpd-serv[216]: %PKT INFRA-FM-3-FAULT MAJOR:
ALARM MAJOR : FPD-NEED-UPGRADE : DECLARE : 0/7:
RP/0/RP1/CPU0:Aug 2 06:19:53.982 UTC: fpd-serv[216]: %PKT INFRA-FM-3-FAULT MAJOR:
ALARM MAJOR : FPD-NEED-UPGRADE : DECLARE : 0/1:
RP/0/RP1/CPU0:Aug 2 06:19:13.764 UTC: imfpga[343]: ISSU Phase Change Notification
[Phase Cleanup]
RP/0/RP1/CPU0:Aug 2 06:19:19.442 UTC: tmgctrl[444]: ISSU Phase Change Notification
[Phase Completed], Becoming Timing Primary now
RP/0/RP1/CPU0:Aug 2 06:19:19.452 UTC: sdr instmgr[1213]: %PKT INFRA-FM-6-FAULT INFO
: INSTALL-IN-PROGRESS : DECLARE : 0/RP1/CPU0: INSTALL IN PROGRESS Alarm : being DECLARED
for the system
RP/0/RP1/CPU0:Aug 2 06:19:19.519 UTC: issudir[139]: %PKT INFRA-FM-6-FAULT INFO:
ISSU-IN-PROGRESS :CLEAR :0/RP1/CPU0:
RP/0/RP1/CPU0:Aug 2 06:19:19.538 UTC: tmgctr1[444]: ISSU Phase Change Notification
[Phase Completed] Processing done
RP/0/RP1/CPU0:Aug 2 06:19:20.480 UTC: sdr instmgr[1213]:
%INSTALL-INSTMGR-2-OPERATION SUCCESS: Install operation 101 finished successfully
RP/0/RP1/CPU0:IOS# install commit
Mon Aug 2 06:44:07.572 UTC
2021-08-02 06:44:10 Install operation 103 started by root123:
  install commit
2021-08-02 06:44:11 Install operation will continue in the background
RP/0/RP1/CPU0:ISO#0/RP1/ADMIN0:Aug 2 06:44:20.839 UTC: inst mgr[6353]:
%PKT INFRA-FM-6-FAULT INFO : INSTALL-IN-PROGRESS :CLEAR :0/RP1: Calvados
{\tt INSTALL\_IN\_PROGRESS~Alarm~:~being~CLEARED~for~the~system}
RP/0/RP1/CPU0:Aug 2 16:44:20.579 UTC: sdr instmgr[1213]: %PKT INFRA-FM-6-FAULT INFO
: INSTALL-IN-PROGRESS :CLEAR :0/RP1/CPU0: INSTALL IN PROGRESS Alarm : being CLEARED
for the system
2021-08-02 06:44:23 Install operation 103 finished successfully
RP/0/RP1/CPU0:Aug 2 06:44:23.201 UTC: sdr instmgr[1213]:
%INSTALL-INSTMGR-2-OPERATION SUCCESS: Install operation 103 finished successfully
sysadmin-vm:0 RP1# show version
Mon Aug 2 07:24:58.414 UTC+00:00
Cisco IOS XR Admin Software, Version 7.3.2
Copyright (c) 2013-2021 by Cisco Systems, Inc.
Build Information:
Built By
           : ingunawa
              : Thu Jul 29 01:27:41 PDT 2021
Built On
 Build Host
             : iox-ucs-003
Workspace
              : /auto/iox-ucs-003-san2/prod/7.3.2.DT IMAGE/ncs560/ws
             : 7.3.2
Version
Location
             : /opt/cisco/calvados/packages/
             : 7.3.2
Label
System uptime is 1 hour, 53 minutes
sysadmin-vm:0 RP1# exit
Mon Aug 2 07:25:02.260 UTC+00:00
```

```
RP/0/RP1/CPU0:IOS#show version
Mon Aug 2 07:25:04.063 UTC
Cisco IOS XR Software, Version 7.3.2
Copyright (c) 2013-2021 by Cisco Systems, Inc.
Build Information:
Built By
          : ingunawa
             : Thu Jul 29 03:34:42 PDT 2021
 Built On
 Built Host : iox-ucs-003
 Workspace : /auto/iox-ucs-003-san2/prod/7.3.2.DT_IMAGE/ncs560/ws
           : 7.3.2
 Version
            : /opt/cisco/XR/packages/
 Location
 Label
             : 7.3.2
cisco NCS-560 () processor
System uptime is 1 hour 13 minutes
RP/0/RP1/CPU0:IOS #
```

8. Check the IMFPGA FPD Versions



Note

IMFPGA FPDs are not upgraded as part of ISSU, as auto-upgrade is disabled. Based on your network design, you must enable and upgrade the IM FPDs as well as reload the IMs at a time when it has the least impact on traffic.

```
RP/0/RP1/CPU0:IOS\#show\ hw-module\ fpd Mon Aug 2 06:38:52.253 UTC
```

Auto-upgrade:Disabled

naco apgro							Versions
Location	Card type	HWver	FPD device	ATR	Status Ru		Programd
0/1	NCS4200-1T16G-PS	0.0	IMFPGA		NEED UPGD	1.95	1.95
0/3	NCS4200-1T16G-PS	0.0	IMFPGA		NEED UPGD	1.95	1.95
0/4	A900-IMA8Z	0.0	IMFPGA		CURRENT	17.05	17.05
0/7	N560-IMA2C	0.0	IMFPGA		NEED UPGD	4.96	4.96
0/9	A900-IMA8Z	0.0	IMFPGA		CURRENT	17.05	17.05
0/RP0	N560-RSP4-E	0.0	ADM		CURRENT	1.06	1.06
0/RP0	N560-RSP4-E	0.0	IOFPGA		CURRENT	0.67	0.67
0/RP0	N560-RSP4-E	0.0	PRIMARY-BIOS		RLOAD REQ	0.16	0.19
0/RP0	N560-RSP4-E	0.0	SATA		CURRENT	2.10	2.10
0/RP1	N560-RSP4-E	0.0	ADM		CURRENT	1.06	1.06
0/RP1	N560-RSP4-E	0.0	IOFPGA		CURRENT	0.67	0.67
0/RP1	N560-RSP4-E	0.0	PRIMARY-BIOS		RLOAD REQ	0.16	0.19
0/RP1	N560-RSP4-E	0.0	SATA		CURRENT	2.10	2.10

0/FT0 N560-FAN-H 1.0 PSOC CURRENT 2.02 2.02

RP/0/RP1/CPU0:IOS#

9. Verify Standby is Node and NSR ready

```
RP/0/RP1/CPU0:IOS#show redundancy
Mon Aug 2 06:39:44.953 UTC
Redundancy information for node 0/RP1/CPU0:
______
Node 0/RP1/CPU0 is in ACTIVE role
Partner node (0/RP0/CPU0) is in STANDBY role
Standby node in 0/RP0/CPU0 is ready
Standby node in 0/RP0/CPU0 is NSR-ready
Reload and boot info
______
RP reloaded Mon Aug 2 06:11:55 2021: 27 minutes ago
Active node booted Mon Aug 2 06:11:55 2021: 27 minutes ago
Standby node boot Mon Aug 2 06:20:02 2021: 19 minutes ago
Standby node last went not ready Mon Aug 2 06:22:46 2021: 16 minutes ago
Standby node last went ready Mon Aug 2 06:26:47 2021: 12 minutes ago
Standby node last went not NSR-ready Mon Aug 2 06:15:01 2021: 24 minutes ago
Standby node last went NSR-ready Mon Aug 2 06:24:52 2021: 14 minutes ago
There have been 0 switch-overs since reload
Active node reload "CARD SHUTDOWN"
Standby node reload "CARD SHUTDOWN
RP/0/RP1/CPU0:IOS#
```

10. IM FPD auto-upgrade: After ISSU

a. Configure the **fpd auto-upgrade enable** command to auto upgrade FPDs for all IMs



Note

After ISSU is complete for RPs, configure FPD auto upgrade and FPD auto reload.



Note

FPD auto reload affects network traffic. Ensure you schedule the reload of IMs at a time when it has the least impact on traffic.



Note

Do not exit configure exclusive mode until all IM FPGA upgrades are automatically completed. Use a separate VTY session to monitor and check status.

```
RP/0/RP1/CPU0:IOS# configure exclusive
Mon Aug 2 06:45:11.297 UTC
RP/0/RP1/CPU0:IOS(config)# fpd auto-upgrade enable
RP/0/RP0/CPU0:IOS(config)# fpd auto-reload enable
RP/0/RP0/CPU0:IOS# commit
```



Note

During this step, all IMs that need FPD upgrades, are upgraded simultaneously. IMs are reloaded automatically for the versions to reflect.

b. Exit **configure exclusive** mode when IMFPGA upgrades are completed on all IMs.

```
RP/0/RP1/CPU0:ROUTER#RP/0/RP1/CPU0:Jun 29 06:44:28.808 UTC: fpd imfpga[430]:
%INFRA-FPD Driver-1-UPGRADE ALERT : Auto fpd triggered for FPD IMFPGA upgrade @
location 0/5/CPU0 please wait until all fpd upgrades are done
RP/0/RP1/CPU0:Jun 29 06:45:25.105 UTC: fpd-serv[125]: %PKT INFRA-FM-3-FAULT MAJOR
: ALARM MAJOR :FPD-NEED-UPGRADE :DECLARE :0/5:
RP/0/RP1/CPU0:Jun 29 06:48:02.020 UTC: fpd imfpga[430]:
%INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD IMFPGA@0/5 image programming completed with
UPGRADE DONE state Info: [DONE ]
RP/0/RP1/CPU0:Jun 29 06:48:02.032 UTC: fpd imfpga[430]:
%INFRA-FPD Driver-1-UPGRADE ALERT : FPD IMFPGA @location 0/5 upgrade completed.
0/RP1/ADMIN0:Jun 29 06:48:02.473 UTC: shelf mgr[3695]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/5 hardware state going to
Operational
0/RP1/ADMIN0:Jun 29 06:48:02.473 UTC: shelf mgr[3695]:
%INFRA-SHELF_MGR-6-CARD_SW_OPERATIONAL : Card: 0/RP1 software state going to
Operational
0/RP1/ADMIN0:Jun 29 06:48:02.474 UTC: shelf mgr[3695]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/RP1 hardware state going to
Operational
RP/0/RP1/CPU0:Jun 29 06:48:08.741 UTC: fpd-serv[125]:
%INFRA-FPD SERVER-1-UPGRADE ALERT : fpd auto-reload configured on this box. Going
to reload node 0/5/CPU0 to activate newly upgraded FPD(s).
0/RP1/ADMIN0:Jun 29 06:48:19.182 UTC: shelf mgr[3695]: %INFRA-SHELF MGR-4-CARD RELOAD
: Reloading card 0/5
0/RP1/ADMIN0:Jun 29 06:48:19.186 UTC: shelf mgr[3695]: %INFRA-SHELF MGR-6-HW EVENT
 : Rcvd HW event HW_EVENT_RESET, event_reason_str '' for card 0/5
0/RP1/ADMIN0:Jun 29 06:48:21.581 UTC: shelf mgr[3695]: %INFRA-SHELF MGR-6-HW EVENT
: Rcvd HW event HW EVENT POWERED OFF, event reason str 'reset requested by user'
for card 0/5
RP/0/RP1/CPU0:Jun 29 06:48:58.746 UTC: fpd-serv[125]: %PKT INFRA-FM-3-FAULT MAJOR
: ALARM MAJOR :FPD-NEED-UPGRADE :CLEAR :0/5:
0/RP1/ADMIN0:Jun 29 06:49:52.266 UTC: shelf mgr[3695]: %INFRA-SHELF MGR-6-HW EVENT
: Rcvd HW event HW EVENT POWERED OFF, event reason str '' for card 0/5
0/RP1/ADMIN0:Jun 29 06:49:57.586 UTC: shelf_mgr[3695]: %INFRA-SHELF_MGR-6-HW_EVENT
: Rcvd HW event HW_EVENT_POWERED_ON, event_reason_str '' for card 0/5
0/RP1/ADMIN0:Jun 29 06:50:23.999 UTC: shelf mgr[3695]: %INFRA-SHELF MGR-6-HW EVENT
 : Rcvd HW event HW_EVENT_OK, event_reason_str '' for card 0/5
0/RP1/ADMIN0:Jun 29 06:50:23.999 UTC: shelf mgr[3695]:
%INFRA-SHELF MGR-6-CARD HW OPERATIONAL : Card: 0/5 hardware state going to
Operational
```

11. Break Reload Actions in Multiple Steps to Avoid Traffic Loss

a. Reload Standby RP

In the following example, RP1 is the active and RP0 is the standby route processor. After the standby RP reloads and boots up, the RP FPDs come to node ready and NSR ready state.



Note

It is expected behavior to see different versions of the RPs in System Admin VM and XR VM.

```
RP/0/RP1/CPU0:IOS# admin
Mon Aug 2 07:34:26.615 UTC
Last login: Mon Aug 2 07:20:24 2021 from 192.0.4.4

cafyauto connected from 192.0.4.4 using ssh on sysadmin-vm:0_RP1
sysadmin-vm:0_RP1# hw-module location 0/RP0 reload
Mon Aug 2 07:34:47.432 UTC+00:00
Reload hardware module ? [no,yes] yes
0/RP1/ADMIN0:Aug 2 07:34:48.728 UTC: shelf mgr[4032]: %INFRA-SHELF MGR-6-USER ACTION
```

: User cafyauto(192.0.4.4) requested CLI action 'graceful card reload' for location 0/RPO

result Card graceful reload request on O/RPO succeeded.

RP/0/RP1/CPU0:Aug 2 07:35:01.263 UTC: fpd-serv[216]: %PKT_INFRA-FM-3-FAULT_MAJOR: ALARM MAJOR: FPD-NEED-UPGRADE: CLEAR: 0/RP0:

RP/0/RP1/CPU0:IOS#show hw-module fpd

Auto-upgrade: Enabled

FPD Versions

Location	==== Card type	HWver	FPD device	ATR	Status	Running			
Programd									
	NCS4200-1T16G-PS								
1.10									
	NCS4200-1T16G-PS	0.0	IMFPGA		RLOAD REQ	1.95			
1.10 0/4	A900-IMA8Z	0.0	IMFPGA		CHDDENE	17 05			
17.05	A900-IMA8Z	0.0	IMF PGA		CURRENT	17.05			
	N560-IMA2C	0.0	IMFPGA		RLOAD REO	4.96			
5.01									
0/9	A900-IMA8Z	0.0	IMFPGA		CURRENT	17.05			
17.05									
	N560-RSP4-E	0.0	ADM		CURRENT	1.06			
1.06	N560-RSP4-E	0.0	IOFPGA		CURRENT	0.67			
0.67	NJ00-RSP4-E	0.0	IOFFGA		CURRENT	0.07			
	N560-RSP4-E	0.0	PRIMARY-BIOS		CURRENT	0.19			
0.19									
0/RP0	N560-RSP4-E	0.0	SATA		CURRENT	2.10			
2.10									
	N560-RSP4-E	0.0	ADM		CURRENT	1.06			
1.06	N560-RSP4-E	0.0	TOEDCA		CHDDENE	0.67			
0.67	N56U-RSP4-E	0.0	IOFPGA		CURRENT	0.67			
	N560-RSP4-E	0.0	PRIMARY-BIOS		RLOAD REO	0.16			
0.19						**-*			
0/RP1	N560-RSP4-E	0.0	SATA		CURRENT	2.10			
2.10									
	N560-FAN-H	1.0	PSOC		CURRENT	2.02			
2.02	anio toal								
RP/0/RP1/CPU0:IOS#									

b. Redundancy Switchover—Swap to the Updated RP as Active RP

After the standby RP is reloaded and booted up, perform the redundancy switchover to this RP with upgraded FPD, as active RP.



Note

Follow Step 11a to upgrade and reload the new standby RP.

RP/0/RP1/CPU0:IOS#redundancy switchover Mon Jun 28 17:16:44.914 UTC

Proceed with switchover 0/RP0/CPU0 -> 0/RP1/CPU0? [confirm]

RP/0/RP1/CPU0:Jun 28 17:16:46.157 UTC: $rmf_svr[234]: %HA-REDCON-4-FAILOVER_REQUESTED$: failover has been requested by operator, waiting to initiate Initiating switch-over.



Note

You can either perform steps 11a and 11b, or you can perform step 11c.

Reload the Router to Reflect the Latest FPD Versions



Note

Reloading the router impacts network traffic.

```
RP/0/RP1/CPU0:IOS#admin
Wed Sep 8 06:10:48.761 UTC
Last login: Wed Sep 8 06:05:30 2021 from 192.0.4.6

root connected from 192.0.4.6 using ssh on sysadmin-vm:0_RP1
sysadmin-vm:0_RP1# hw-module location all reload
Wed Sep 8 06:10:56.203 UTC+00:00
Reloading the module will be traffic impacting if not properly drained. Continue
to Reload hardware module ? [no,yes] yes
result Card graceful reload request on all acknowledged.
sysadmin-vm:0 RP1#
```

Automatic FPD Upgrade for PSU

During the installation and Power Supply Unit (PSU) insertion process, the Field-Programmable Devices (FPD) associated with the PSUs are automatically upgraded.



Note

The PSUs are upgraded sequentially, hence the PSU FPD upgrades take longer. You can choose to exclude PSUs from the auto upgrade flow. This restricts the PSUs from being upgraded either upon insertion, or during system upgrade.

To exclude the PSU FPDs from auto upgrading, use the following CLI:

fpd auto-upgrade exclude pm

```
RP/0/RSP0/CPU0:router# show running-config fpd auto-upgrade Wed Mar 30 20:52:55.079 UTC fpd auto-upgrade enable fpd auto-upgrade exclude pm
```



Note

When you upgrade from an earlier unsupported version to a version that supports Automatic FPD upgrade for PSU, the PSU upgrade might happen on bootup.

Upgrade Failure

On failure of an FPD upgrade, you get a warning with the following syslog message:

LC/0/5/CPU0:Jun 27 05:02:25.742 UTC: optics_driver[216]: %INFRA-FPD_Driver-1-UPGRADE_ALERT: FPD MIFPGA@0/5 image programming completed with UPGD FAIL state Info: [Image verification failed at offset 0x5c8, flash value = 0x0, image value = 0x40, image size = 4194304]

LC/0/5/CPU0:Jun 27 05:02:26.570 UTC: optics_driver[216]: %INFRA-FPD_Driver-1-UPGRADE_ALERT : FPD MIFPGA@0/5 image programming completed with UPGD FAIL state Info: [Image verification failed at offset 0x1e, flash value = 0x56, image value = 0xff, image size = 4194304]

When you use the **show hw-module fpd**command, the status column displays **UPGD FAIL** to indicate failure of the FPD upgrade.



Note

- Do not reload the line card with a failed FPD upgrade image.
- Upgrade failed FPDs will be fixed with a manual upgrade.
- Contact Cisco TAC or your account representative if the FPD upgrade failure is not repaired.

Upgrade Failure