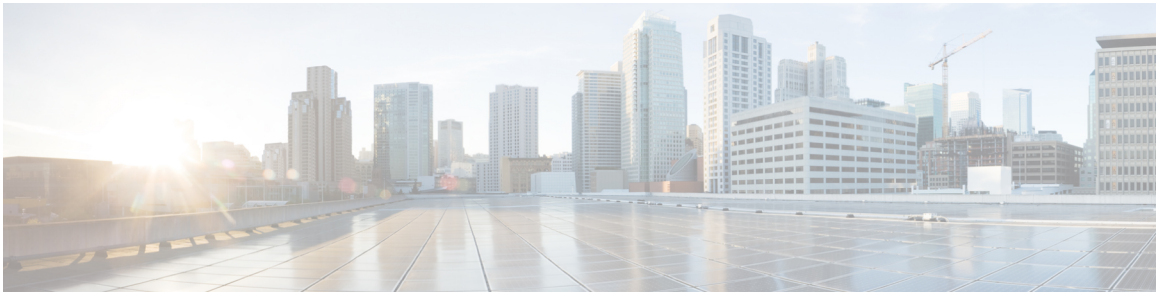
Network Services Configuration Guide, Cisco IOS XE 17.x

First Published: 2022-11-17

Last Modified: 2024-03-27

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



CONTENTS

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

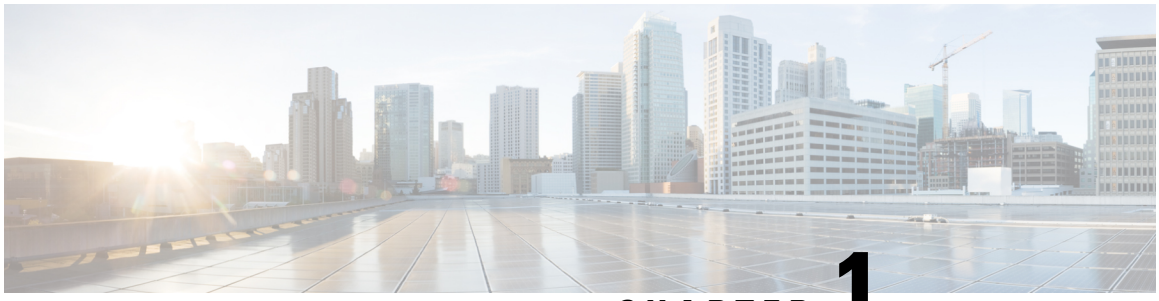
Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.

PART I

Cisco Networking Services



CHAPTER 1

Configuring Cisco Networking Services

The Cisco Networking Services (CNS) feature is a collection of services that can provide remote event-driven configuring of Cisco IOS networking devices and remote execution of some command-line interface (CLI) commands.

Prerequisites for Cisco Networking Services

- Configure the remote device to support the Cisco Networking Services configuration agent and the Cisco Networking Services event agent.
- Configure a transport protocol on the remote device that is compatible with the remote device's external interface. The following table lists the supported transport protocols that can be used depending on the device interface.
- Create the configuration template in the Cisco Networking Services configuration-engine provisioning database. (This task is best done by a senior network designer.)

Table 1: Device Interface and Transport Protocols Required by Cisco Networking Services Services

Device Interface	SLARP Transport Protocol	ATM InARP Transport Protocol	PPP (IPCP) Transport Protocol
T1	Yes	Yes	Yes
ADSL	No	Yes	Yes
Serial	Yes	No	Yes

Restrictions for Cisco Networking Services

Cisco Networking Services Configuration Engine

- The Cisco Networking Services configuration engine must be the Cisco Intelligence Engine 2100 (Cisco IE2100) series and must be running software version 1.3.
- The configuration engine must have access to an information database of attributes for building a configuration. This database can reside on the Cisco IE2100 itself.
- Configuration templates must be prepared on the Cisco Networking Services configuration engine before installation of the remote device.
- The user of Cisco Networking Services Flow-Through Provisioning and the Cisco Networking Services configuration engine must be familiar with designing network topologies, designing configuration templates, and using the Cisco Networking Services configuration engine.

Remote Device

- The remote device must run a Cisco IOS image that supports the Cisco Networking Services configuration agent and Cisco Networking Services event agent.
- Ports must be prepared on the remote device for connection to the network.
- You must ensure that the remote device is configured using Cisco Configuration Express.

Information About Cisco Networking Services

Cisco Networking Services

Cisco Networking Services is a foundation technology for linking users to networking services and provides the infrastructure for the automated configuration of large numbers of network devices. Many IP networks are complex with many devices, and each device must currently be configured individually. When standard configurations do not exist or have been modified, the time involved in initial installation and subsequent upgrading is considerable. The volume of smaller, more standardized, customer networks is also growing faster than the number of available network engineers. Internet service providers (ISPs) now need a method for sending out partial configurations to introduce new services. To address all these issues, Cisco Networking Services has been designed to provide “plug-and-play” network services using a central directory service and distributed agents. Cisco Networking Services features include Cisco Networking Services configuration and event agents and a Flow-Through Provisioning structure. The configuration and event agents use a Cisco Networking Services configuration engine to provide methods for automating initial Cisco device configurations, incremental configurations, and synchronized configuration updates, and the configuration engine reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe. The Cisco Networking Services Flow-Through Provisioning uses the Cisco Networking Services configuration and event agents to provide an automated workflow, eliminating the need for an on-site technician.

Cisco Networking Services EXEC Agent

The CNS EXEC agent allows a remote application to execute an EXEC mode CLI command on a Cisco device by sending an event message that contains the command. A restricted set of EXEC **show** commands is supported.

Cisco Networking Services Results Messages

When a partial configuration has been received by the device, each line of the configuration will be applied in the same order as it was received. If the Cisco parser has an error with one of the lines of the configuration, then all the configuration up to this point will be applied to the device, but none of the configuration beyond the error will be applied. If an error occurs, the **cns config partial** command will retry until the configuration successfully completes. In the pull mode, the command will not retry after an error. By default, NVRAM will be updated except when the **no-persist** keyword is configured.

A message will be published on the Cisco Networking Services event bus after the partial configuration is complete. The Cisco Networking Services event bus will display one of the following status messages:

- cisco.mgmt.cns.config.complete—Cisco Networking Services configuration agent successfully applied the partial configuration.
- cisco.mgmt.cns.config.warning—Cisco Networking Services configuration agent fully applied the partial configuration, but encountered possible semantic errors.
- cisco.mgmt.cns.config.failure (CLI syntax)—Cisco Networking Services configuration agent encountered a command line interface (CLI) syntax error and was not able to apply the partial configuration.
- cisco.mgmt.cns.config.failure (CLI semantic)—Cisco Networking Services configuration agent encountered a CLI semantic error and was not able to apply the partial configuration.

With the CNS Enhanced Results Messages feature, a second message is sent to the subject "cisco.cns.config.results" in addition to the appropriate message above. The second message contains both overall and line-by-line information about the configuration that was sent and the result of the action requested in the original message. If the action requested was to apply the configuration, then the information in the results message is semantic in nature. If the action requested was to check syntax only, then the information in the results message is syntactical in nature.

Cisco Networking Services Message Formats

SOAP Message Format

Using the Service-Oriented Access Protocol (SOAP) protocol provides a way to format the layout of Cisco Networking Services messages in a consistent manner. SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. SOAP uses extensible markup language (XML) technologies to define an extensible messaging framework that provides a message format that can be exchanged over a variety of underlying protocols.

Within the SOAP message structure, there is a security header that enables Cisco Networking Services notification messages to authenticate user credentials.

Cisco Networking Services messages are classified into three message types: request, response and notification. The formats of these three message types are defined below.

Request Message

The following is the format of a Cisco Networking Services request message to the Cisco device:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope">
  <SOAP:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="0">
      <wsse:usernameToken>
        <wsse:Username>john</wsse:Username>
        <wsse:Password>cisco</wsse:Password>
      </wsse:usernameToken>
    </wsse:Security>
    <cns:cnsHeader version="1.0" xmlns:cns="http://www.cisco.com/management/cns/envelope">
      <cns:Agent>CNS_CONFIG</cns:Agent>
      <cns:Request>
        <cns:correlationID>IDENTIFIER</cns:correlationID>
        <cns:ReplyTo>
          <cns:URL>http://10.1.36.9:80/cns/ResToServer</cns:URL>
        </cns:ReplyTo>
      </cns:Request>
      <cns:Time>2003-04-23T20:27:19.847Z</cns:Time>
    </cns:cnsHeader>
  </SOAP:Header>
  <SOAP:Body xmlns="http://www.cisco.com/management/cns/config">
    <config-event config-action="read" no-syntax-check="TRUE">
      <config-data>
        <config-id>AAA</config-id>
        <cli>access-list 1 permit any</cli>
      </config-data>
    </config-event>
  </SOAP:Body>
</SOAP:Envelope>
```

Note The ReplyTo field is optional. In the absence of the ReplyTo field, the response to the request will be sent to the destination where the request originated. The body portion of this message contains the payload and is processed by the Cisco Networking Services agent mentioned in the Agent field.

Response Message

The following is the format of a Cisco Networking Services response message from the Cisco device as a response to a request:

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username infysj-7204-8 /wsse:Username
wsse:Password NTM3NTg2NzIzOTg2MTk2MjgzNQ==/wsse:Password
/wsse:UsernameToken /wsse:Security
CNS:cnsHeader Version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG /CNS:Agent
CNS:Response
CNS:correlationID IDENTIFIER /CNS:correlationID
/CNS:Response
```

```
CNS:Time 2005-06-23T16:27:36.185Z /CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config"
config-success config-id AAA /config-id /config-success
/SOAP:Body
/SOAP:Envelope
```

Note The value of CorrelationId is echoed from the corresponding request message.

The body portion of this message contains the response from the Cisco device to a request. If the request is successfully processed, the body portion contains the value of the response put in by the agent that processed the request. If the request cannot be successfully processed, then the body portion will contain an error response.

Notification Message

The following is the format of a Cisco Networking Services notification message sent from the Cisco device:

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username dvlpr-7200-2 /wsse:Username
wsse:Password /wsse:Password
/wsse:UsernameToken
/wsse:Security
CNS:cnsHeader version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG_CHANGE/CNS:Agent
CNS:Notify /CNS:Notify
CNS:Time 2006-01-09T18:57:08.441Z/CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config-change"
configChanged version="1.1" sessionData="complete"
sequence lastReset="2005-12-11T20:18:39.673Z" 7 /sequence
changeInfo
user/user
async port con_0 /port /async
when
absoluteTime 2006-01-09T18:57:07.973Z /absoluteTime
/when
/changeInfo
changeData
changeItem
context /context
enteredCommand
cli access-list 2 permit any /cli
/enteredCommand
oldConfigState
cli access-list 1 permit any /cli
/oldConfigState
newConfigState
cli access-list 1 permit any /cli
cli access-list 2 permit any /cli
/newConfigState
/changeItem
/changeData
```

```
/configChanged
/SOAP:Body
/SOAP:Envelope
```

A notification message is sent from the Cisco device without a corresponding request message when a configuration change is made. The body of the message contains the payload of the notification and it may also contain error information. If the request message sent to the Cisco device fails in XML parsing and the CorrelationId field cannot be parsed, then an error notification message will be sent instead of an error response.

Error Reporting

Error is reported in the body of the response or a notification message in the SOAP Fault element. The following is the format for reporting errors.

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username dvlpr-7200-2 /wsse:Username
wsse:Password /wsse:Password
/wsse:UsernameToken
/wsse:Security
CNS:cnsHeader version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG /CNS:Agent
CNS:Response
CNS:correlationID SOAP_IDENTIFIER /CNS:correlationID
/CNS:Response
CNS:Time 2006-01-09T19:10:10.009Z /CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config"
SOAP:Detail
config-failure
config-id AAA /config-id
error-info
line-number 1 /line-number
error-message CNS_INVALID_CLI_CMD /error-message
/error-info
/config-failure
/SOAP:Detail
/SOAP:Fault
/SOAP:Body
/SOAP:Envelope
```

Cisco Networking Services IDs

The Cisco Networking Services ID is a text string that is used exclusively with a particular Cisco Networking Services agent. The Cisco Networking Services ID is used by the Cisco Networking Services agent to identify itself to the server application with which it communicates. For example, the Cisco Networking Services configuration agent will include the configuration ID when communicating between the networking device and the configuration server. The configuration server uses the Cisco Networking Services configuration ID as a key to locate the attribute containing the Cisco CLI configuration intended for the device that originated the configuration pull.

The network administrator must ensure a match between the Cisco Networking Services agent ID as defined on the routing device and the Cisco Networking Services agent ID contained in the directory attribute that

corresponds to the configuration intended for the routing device. Within the routing device, the default value of the Cisco Networking Services agent ID is always set to the hostname. If the hostname changes, the Cisco Networking Services agent ID also changes. If the Cisco Networking Services agent ID is set using the CLI, any change will be followed by a message sent to syslog or an event message will be sent.

The Cisco Networking Services agent ID does not address security issues.

Cisco Networking Services Password

The Cisco Networking Services password is used to authenticate the Cisco Networking Services device. You must configure the Cisco Networking Services password the first time a device is deployed, and the Cisco Networking Services password must be the same as the bootstrap password set on the Configuration Engine (CE). If both the device and the CE bootstrap password use their default settings, a newly deployed device will be able to connect to the CE. Once connected, the CE manages the Cisco Networking Services password. Network administrators must ensure not to change the Cisco Networking Services password. If the Cisco Networking Services password is changed, connectivity to the CE will be lost.

Cisco Networking Services Zero Touch

The Cisco Networking Services Zero Touch feature provides a zero touch deployment solution where the device contacts a Cisco Networking Services configuration engine to retrieve its full configuration automatically. This capability is made possible through a single generic bootstrap configuration file common across all service provider end customers subscribing to the services. Within the Cisco Networking Services framework, customers can create this generic bootstrap configuration without device-specific or network-specific information such as interface type, line type, or controller type (if applicable).

The Cisco Networking Services connect functionality is configured with a set of Cisco Networking Services connect templates. A Cisco Networking Services connect profile is created for connecting to the Cisco Networking Services configuration engine and to implement the Cisco Networking Services connect templates on a Customer Premise Equipment (CPE) device. Cisco Networking Services connect variables can be used as placeholders within a Cisco Networking Services connect template configuration. These variables, such as the active DLCI, are substituted with real values before the Cisco Networking Services connect templates are sent to the device's parser.

To use the zero touch functionality, the device that is to be initialized must have a generic bootstrap configuration. This configuration includes Cisco Networking Services connect templates, Cisco Networking Services connect profiles, and the **cns config initial** command. This command initiates the Cisco Networking Services connect function.

The Cisco Networking Services connect functionality performs multiple ping iterations through the device's interfaces and lines, as well as any available controllers. For each iteration, the Cisco Networking Services connect function attempts to ping the Cisco Networking Services configuration engine. If the ping is successful, the pertinent configuration information can be downloaded from the Cisco Networking Services configuration engine. If connectivity to the Cisco Networking Services configuration engine is unsuccessful, the Cisco Networking Services connect function removes the configuration applied to the selected interface, and the Cisco Networking Services connect process restarts with the next available interface specified by the Cisco Networking Services connect profile.

The Cisco Networking Services Zero Touch feature provides the following benefits:

- Ensures consistent Cisco Networking Services commands.
- Use of a channel service unit (E1 or T1 controller) is allowed.

How to Configure Cisco Networking Services

Deploying the Cisco Networking Services Device

Incremental or partial configuration allows the remote device to be incrementally configured after its initial configuration. You must perform these configurations manually through the Cisco Networking Services configuration engine. The registrar allows you to change the configuration templates, edit parameters, and submit the new configuration to the device without a software or hardware restart.

Before you begin

Perform this task to manually install an initial Cisco Networking Services configuration.

Your remote device arrives from the factory with a bootstrap configuration. Upon initial power-on, the device automatically pulls a full initial configuration from the Cisco Networking Services configuration engine, although you can optionally arrange for this manually as well. After initial configuration, you can optionally arrange for periodic incremental (partial) configurations for synchronization purposes.

For more details on using the Cisco CNS configuration engine to automatically install the initial CNS configuration, see the *Cisco CNS Configuration Engine Administrator's Guide* at http://www.cisco.com/en/US/docs/net_mgmt/configuration_engine/1.3/administration/guide/ag13.html

Initial Cisco Networking Services Configuration

Initial configuration of the remote device occurs automatically when the device is initialized on the network. Optionally, you can perform this configuration manually.

Cisco Networking Services assigns the remote device a unique IP address or hostname. After resolving the IP address (using Serial Line Address Resolution Protocol (SLARP), ATM Inverse ARP (ATM InARP), or PPP protocols), the system optionally uses Domain Name System (DNS) reverse lookup to assign a hostname to the device and invokes the Cisco Networking Services agent to download the initial configuration from the Cisco Networking Services configuration engine.

Incremental Configuration

Before you can configure an incremental configuration, Cisco Networking Services must be operational and the required Cisco Networking Services agents configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns template connect** *name*
4. **cli** *config-text*
5. Repeat Step 4 to add all required CLI commands.
6. **exit**
7. **cns connect** *name* [**retry-interval** *interval-seconds*] [**retries** *number-retries*] [**timeout** *timeout-seconds*] [**sleep** *sleep-seconds*]
8. Do one of the following:
   - **discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
   - **template** *name*

9. **exit**
10. **cns config initial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**]
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cns template connect** *name* **Example:** `Device(config)# cns template connect template 1`	Enters Cisco Networking Services template connect configuration mode and defines the name of a Cisco Networking Services connect template.
Step 4	**cli** *config-text* **Example:** `Device(config-templ-conn)# cli encapsulation ppp`	Specifies commands to configure the interface.
Step 5	Repeat Step 4 to add all required CLI commands. **Example:** `Device(config-templ-conn)# cli ip directed-broadcast`	Repeat Step 4 to add other CLI commands to configure the interface or to configure the modem lines.
Step 6	**exit** **Example:** `Device(config-templ-conn)# exit`	Exits Cisco Networking Services template connect configuration mode and completes the configuration of a Cisco Networking Services connect template. **Note** Entering the **exit** command is required. This requirement was implemented to prevent accidentally entering a command without the **cli** command.
Step 7	**cns connect** *name* [**retry-interval** *interval-seconds*] [**retries** *number-retries*] [**timeout** *timeout-seconds*] [**sleep** *sleep-seconds*] **Example:**	Enters Cisco Networking Services connect configuration mode and defines the parameters of a Cisco Networking Services connect profile for connecting to the Cisco Networking Services configuration engine.

	Command or Action	Purpose
	`Device(config)# cns connect profile-1 retry-interval 15 timeout 90`	

Step 8	Do one of the following:
	• **discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
	• **template** *name*
	Example:
	`Device(config-cns-conn)# discover interface serial`
	Example:
	`Device(config-cns-conn)# template template-1`
		(Optional) Configures a generic bootstrap configuration.
		• **discover** —Defines the interface parameters within a Cisco Networking Services connect profile for connecting to the Cisco Networking Services configuration engine.
		or
		• **template** —Specifies a list of Cisco Networking Services connect templates within a Cisco Networking Services connect profile to be applied to a device's configuration.

Step 9	**exit** **Example:** `Device(config-cns-conn)# exit`	Exits Cisco Networking Services connect configuration mode and returns to global configuration mode.
Step 10	**cns config initial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**] **Example:** `Device(config)# cns config initial 10.1.1.1 no-persist`	Starts the Cisco Networking Services configuration agent, connects to the Cisco Networking Services configuration engine, and initiates an initial configuration. You can use this command only before the system boots for the first time. **Note** The optional **encrypt** keyword is available only in images that support Secure Socket Layer (SSL). **Caution** If you write the new configuration to NVRAM by omitting the **no-persist**keyword, the original bootstrap configuration is overwritten.
Step 11	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Advanced Cisco Networking Services Features

Perform this task to configure more advanced Cisco Networking Services features. After the Cisco Networking Services agents are operational, you can configure some other features. You can enable the Cisco Networking Services inventory agent--that is, send an inventory of the device's line cards and modules to the Cisco Networking Services configuration engine--and enter Cisco Networking Services inventory mode.

Some other advanced features allow you to use the Software Developer's Toolkit (SDK) to specify how Cisco Networking Services notifications should be sent or how to access MIB information. Two encapsulation methods can be used: either nongranular (SNMP) encapsulation or granular (XML) encapsulation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns mib-access encapsulation** {**snmp** | **xml**[**size** *bytes*]}
4. **cns notifications encapsulation** {**snmp** | **xml**}
5. **cns inventory**
6. **transport event**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cns mib-access encapsulation** {**snmp** | **xml**[**size** *bytes*]} **Example:** `Device(config)# cns mib-access encapsulation snmp`	(Optional) Specifies the type of encapsulation to use when accessing MIB information. • Use the **snmp** keyword to specify that nongranular encapsulation is used to access MIB information. • Use the **xml** keyword to specify that granular encapsulation is used to access MIB information. The optional **size** keyword specifies the maximum size for response events, in bytes. The default byte value is 3072.
Step 4	**cns notifications encapsulation** {**snmp** | **xml**} **Example:** `Device(config)# cns notifications encapsulation xml`	(Optional) Specifies the type of encapsulation to use when sending Cisco Networking Services notifications. • Use the **snmp** keyword to specify that nongranular encapsulation is used when Cisco Networking Services notifications are sent. • Use the **xml** keyword to specify that granular encapsulation is used when Cisco Networking Services notifications are sent.

	Command or Action	Purpose
Step 5	**cns inventory** **Example:** `Device(config)# cns inventory`	Enables the Cisco Networking Services inventory agent and enters Cisco Networking Services inventory mode. • An inventory of the device's line cards and modules is sent to the Cisco Networking Services configuration engine.
Step 6	**transport event** **Example:** `Device(cns-inv)# transport event`	Specifies that inventory requests are sent out with each Cisco Networking Services inventory agent message.
Step 7	**exit** **Example:** `Device(cns-inv)# exit`	Exits Cisco Networking Services inventory mode and returns to global configuration mode. • Repeat this command to return to privileged EXEC mode.

Troubleshooting Cisco Networking Services Agents

This section explains how to troubleshoot Cisco Networking Services agent issues.

The **show** commands created for the Cisco Networking Services image agent display information that is reset to zero after a successful reload of the device. Depending on the configuration of the image distribution process, the new image may not reload immediately. When a reload is not immediate or has failed, use the Cisco Networking Services image agent **show** commands to determine whether the image agent has connected to the image distribution server over HTTP or whether the image agent is receiving events from an application over the Cisco Networking Services Event Bus.

SUMMARY STEPS

1. **enable**
2. **show cns image status**
3. **clear cns image status**
4. **show cns image connections**
5. **show cns image inventory**
6. **debug cns image** [**agent**| **all**| **connection**| **error**]
7. **show cns event connections**
8. **show cns event subject** [*name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show cns image status** **Example:** `Device# show cns image status`	(Optional) Displays information about the Cisco Networking Services image agent status.
Step 3	**clear cns image status** **Example:** `Device# clear cns image status`	(Optional) Clears Cisco Networking Services image agent status statistics.
Step 4	**show cns image connections** **Example:** `Device# show cns image connections`	(Optional) Displays information about Cisco Networking Services image management server HTTP or HTTPS connections.
Step 5	**show cns image inventory** **Example:** `Device# show cns image inventory`	(Optional) Displays inventory information about the Cisco Networking Services image agent. • This command displays a dump of XML that would be sent out in response to an image agent inventory request message. The XML output can be used to determine the information requested by an application.
Step 6	**debug cns image** [**agent**| **all**| **connection**| **error**] **Example:** `Device# debug cns image all`	(Optional) Displays debugging messages for Cisco Networking Services image agent services.
Step 7	**show cns event connections** **Example:** `Device# show cns event connections`	(Optional) Displays the status of the Cisco Networking Services event agent connection--such as whether it is connecting to the gateway, connected, or active--and to display the gateway used by the event agent and its IP address and port number.
Step 8	**show cns event subject** [*name*] **Example:** `Device# show cns event subject subject1`	(Optional) Displays a list of subjects of the Cisco Networking Services event agent that are subscribed to by applications.

Examples

In the following example, status information about the Cisco Networking Services image agent is displayed using the **show cns image status**privileged EXEC command:

```
Device# show cns image status
Last upgrade started at 11:45:02.000 UTC Mon May 6 2003
Last upgrade ended at 11:56:04.000 UTC Mon May 6 2003 status SUCCESS
Last successful upgrade ended at 11:56:04.000 UTC Mon May 6 2003
```

```
Last failed upgrade ended at 06:32:15.000 UTC Wed Apr 16 2003
Number of failed upgrades: 2
Number of successful upgrades: 6
 messages received: 12
 receive errors: 5
Transmit Status
  TX Attempts:4
    Successes:3         Failures 2
```

In the following example, information about the status of the Cisco Networking Services image management HTTP connections is displayed using the **show cns image connections**privileged EXEC command:

```
show cns image connections
CNS Image Agent:  HTTP connections
Connection attempts 1
never connected:0   Abrupt disconnect:0
Last successful connection at 11:45:02.000 UTC Mon May 6 2003
```

In the following example, information about the Cisco Networking Services image agent inventory is displayed using the **show cns image inventory**privileged EXEC command:

```
show cns image inventory
Inventory Report
imageInventoryReport deviceName imageID Router /imageID hostName Router /ho
IOS (tm) C2600 Software (C2600-I-M), Experimental Version 12.3(20030414:081500)]
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Mon 14-Apr-03 02:03 by engineer /versionString imageFile tftp://10.25.2.1.
```

In the following example, debugging messages for all Cisco Networking Services image agent services are displayed using the **debug cns image**privileged EXEC command. The Cisco Networking Services image agent in this example is connecting to an image server over HTTP. After connecting, the image server asks for an inventory of the Cisco device.

```
Device# debug cns image all
All cns image debug flags are on
Device# cns image retrieve

May  7 06:11:42.175: CNS Image Agent: set EXEC lock
May  7 06:11:42.175: CNS Image Agent: received message from EXEC
May  7 06:11:42.175: CNS Image Agent: set session lock 1
May  7 06:11:42.175: CNS Image Agent: attempting to send to
destination(http://10.1.36.8:8080/imgsrv/xgate):
?xml version="1.0" encoding="UTF-8"? cnsMessageversion="1.0" senderCredentials userName
dvlpr-7200-6 /userName /senderCredentials
messageID dvlpr-7200-6_2 /messageID sessionControl imageSessionStart version="1.0"
initiatorInfotrigger EXEC/trigger initiatorCredentials userName dvlpr-7200-6/userName
/initiatorCredentials /initiatorInfo /imageSessionStart /sessionControl /cnsMessage
May  7 06:11:42.175: CNS Image Agent: clear EXEC lock
May  7 06:11:42.175: CNS Image Agent: HTTP message sent url:http://10.1.36.8:8080/imgsrv/xgate
May  7 06:11:42.191: CNS Image Agent: response data alloc 4096 bytes
May  7 06:11:42.191: CNS Image Agent: HTTP req data free
May  7 06:11:42.191: CNS Image Agent: response data freed
May  7 06:11:42.191: CNS Image Agent: receive message
?xml version="1.0" encoding="UTF-8"?
cnsMessage version="1.0"
senderCredentials
userName myImageServer.cisco.com/userName
passWord R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b/passWord
/senderCredentials
messageID dvlpr-c2600-2-476456/messageID
```

```
request
replyTo
serverReply http://10.1.36.8:8080/imgsrv/xgate /serverReply
/replyTo
imageInventory
inventoryItemList
all/
/inventoryItemList
/imageInventory
/request
/cnsMessage
```

The following example displays the IP address and port number of the primary and backup gateways:

```
Device# show cns event connections
The currently configured primary event gateway:
 hostname is 10.1.1.1.
 port number is 11011.
Event-Id is Internal test1
Keepalive setting:
 none.
Connection status:
 Connection Established.
The currently configured backup event gateway:
 none.
The currently connected event gateway:
 hostname is 10.1.1.1.
 port number is 11011.
```

The following sample displays a list of subjects of the Cisco Networking Services event agent that are subscribed to by applications:

```
Device# show cns event subject
The list of subjects subscribed by applications.
 cisco.cns.mibaccess:request
 cisco.cns.config.load
 cisco.cns.config.reboot
 cisco.cns.exec.cmd
```

Configuration Examples for Cisco Networking Services

Example: Deploying the Cisco Networking Services Device

The following example shows an initial configuration on a remote device. The hostname of the remote device is the unique ID. The Cisco Networking Services configuration engine IP address is 172.28.129.22.

```
cns template connect template1
 cli ip address negotiated
 cli encapsulation ppp
 cli ip directed-broadcast
 cli no keepalive
 cli no shutdown
 exit
cns connect host1 retry-interval 30 retries 3
exit
 hostname RemoteRouter
 ip route 172.28.129.22 255.255.255.0 10.11.11.1
```

```
cns id Ethernet 0 ipaddress
cns config initial 10.1.1.1 no-persist
exit
```

Example: Using the Cisco Networking Services Zero Touch Solution

Configuring PPP on a Serial Interface

The following example shows the bootstrap configuration for configuring PPP on a serial interface:

```
cns template connect ppp-serial
cli ip address negotiated
cli encapsulation ppp
cli ip directed-broadcast
cli no keepalive
exit
cns template connect ip-route
cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect serial-ppp ping-interval 1 retries 1
discover interface serial
template ppp-serial
template ip-route
exit
hostname 26ML
cns config initial 10.1.1.1 no-persist inventory
```

Configuring PPP on an Asynchronous Interface

The following example shows the bootstrap configuration for configuring PPP on an asynchronous interface:

```
cns template connect async
cli modem InOut
 .
 .
 .
exit
cns template connect async-interface
cli encapsulation ppp
cli ip unnumbered FastEthernet0/0
cli dialer rotary-group 0
exit
cns template connect ip-route
cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect async
discover line Async
template async
discover interface
template async-interface
template ip-route
exit
hostname async-example
cns config initial 10.1.1.1 no-persist inventory
```

Configuring HDLC on a Serial Interface

The following example shows the bootstrap configuration for configuring High-Level Data Link Control (HDLC) on a serial interface:

```
cns template connect hdlc-serial
cli ip address slarp retry 1
exit
cns template connect ip-route
cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect hdlc-serial ping-interval 1 retries 1
discover interface serial
template hdlc-serial
template ip-route
exit
hostname host1
cns config initial 10.1.1.1 no-persist inventory
```

Configuring Aggregator Device Interfaces

The following examples show how to configure a standard serial interface and a serial interface bound to a controller on an aggregator device (also known as the DCE). In order for connectivity to be established, the aggregator device must have a point-to-point subinterface configured.

Standard Serial Interface

```
interface Serial0/1
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
exit
interface Serial0/1.1 point-to-point
 10.0.0.0 255.255.255.0
 frame-relay interface-dlci 8
```

Serial Interface Bound to a Controller

```
controller T1 0
 framing sf
 linecode ami
 channel-group 0 timeslots 1-24
exit
interface Serial0:0
 no ip address
 encapsulation frame-relay
 frame-relay intf-type dce
exit
interface Serial0:0.1 point-to-point
 ip address ip-address mask
 frame-relay interface-dlci dlci
```

Configuring IP over Frame Relay

The following example shows the bootstrap configuration for configuring IP over Frame Relay on a CPE device:

```
cns template connect setup-frame
 cli encapsulation frame-relay
 exit
cns template connect ip-over-frame
 cli frame-relay interface-dlci ${dlci}
 cli ip address dynamic
```

```
 exit
cns template connect ip-route
 cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
 exit
cns connect ip-over-frame
 discover interface Serial
 template setup-frame
 discover dlci
 template ip-over-frame
 template ip-route
exit
cns config initial 10.1.1.1
```

Configuring IP over Frame Relay over T1

The following example shows the bootstrap configuration for configuring IP over Frame Relay over T1 on a CPE device:

```
cns template connect setup-frame
 cli encapsulation frame-relay
 exit
cns template connect ip-over-frame
 cli frame-relay interface-dlci ${dlci}
 cli ip address dynamic
 exit
cns template connect ip-route
 cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
 exit
cns template connect t1-controller
 cli framing esf
 cli linecode b8zs
 cli channel-group 0 timeslots 1-24 speed 56
 exit
cns connect ip-over-frame-over-t1
 discover controller T1
 template t1-controller
 discover interface
 template setup-frame
 discover dlci
 template ip-over-frame
 template ip-route
exit
cns config initial 10.1.1.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	Cisco IOS Cisco Networking Services Command Reference

Related Topic	Document Title
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Standards and RFCs

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

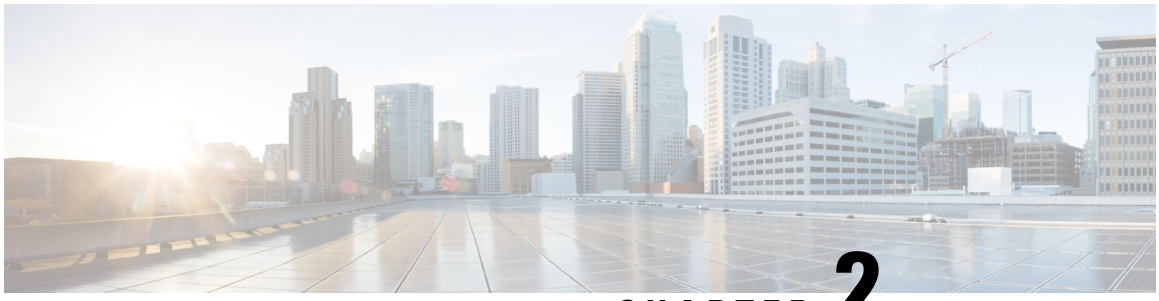
Feature Information for Cisco Networking Services

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Cisco Networking Services

Feature Name	Releases	Feature Information
Cisco Networking Services	Cisco IOS XE Release 2.1 12.2(25)S 12.2(33) SRA 12.2(33)SB 12.2(33)SXI	The Cisco Networking Services feature is a collection of services that can provide remote event-driven configuring of Cisco IOS networking devices and remote execution of some CLI commands. The following commands were introduced or modified by this feature: **clear cns config stats**, **clear cns counters**, **clear cns event stats**, **cli (cns)**, **cns config cancel**, **cns config initial**, **cns config notify**, **cns config partial**, **cns config retrieve**, **cns connect**, **cns event**, **cns exec**, **cns id**, **cns template connect**, **cns trusted-server**, **debug cns config**, **debug cns exec**, **debug cns xml-parser**, **logging cns-events**, **show cns config stats**, **show cns event connections**, **show cns event stats**, **show cns event subject**.



CHAPTER 2

CNS Configuration Agent

Information About CNS Configuration Agent

Cisco Networking Services Configuration Agent

The Cisco Networking Services configuration agent is involved in the initial configuration and subsequent partial configurations on a Cisco device. To activate the Cisco Networking Services configuration agent, enter any of the **cns config** CLI commands.

Initial Cisco Networking Services Configuration

When a routing device first comes up, it connects to the configuration server component of the Cisco Networking Services configuration agent by establishing a TCP connection through the use of the **cns config initial**command, a standard CLI command. The device issues a request and identifies itself by providing a unique configuration ID to the configuration server.

When the Cisco Networking Services web server receives a request for a configuration file, it invokes the Java servlet and executes the corresponding embedded code. The embedded code directs the Cisco Networking Services web server to access the directory server and file system to read the configuration reference for this device (configuration ID) and template. The Configuration Agent prepares an instantiated configuration file by substituting all the parameter values specified in the template with valid values for this device. The configuration server forwards the configuration file to the Cisco Networking Services web server for transmission to the routing device.

The Cisco Networking Services configuration agent accepts the configuration file from the Cisco Networking Services web server, performs XML parsing, checks syntax (optional), and loads the configuration file. The routing device reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe.

For more details on using the Cisco Cisco Networking Services configuration engine to automatically install the initial Cisco Networking Services configuration, see the *Cisco Networking Services Configuration Engine Administrator's Guide* at http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cns/ce/rel13/ag13/index.htm .

Incremental Cisco Networking Services Configuration

Once the network is up and running, new services can be added using the Cisco Networking Services configuration agent. Incremental (partial) configurations can be sent to routing devices. The actual configuration can be sent as an event payload by way of the event gateway (push operation) or as a signal event that triggers the device to initiate a pull operation.

The routing device can check the syntax of the configuration before applying it. If the syntax is correct, the routing device applies the incremental configuration and publishes an event that signals success to the configuration server. If the device fails to apply the incremental configuration, it publishes an event that indicates an error.

Once the routing device has applied the incremental configuration, it can write the configuration to NVRAM or wait until signaled to do so.

Synchronized Configuration

When a routing device receives a configuration, the device has the option to defer application of the configuration upon receipt of a write-signal event. The Cisco Networking Services Configuration Agent feature allows the device configuration to be synchronized with other dependent network activities.

How to Configure CNS Configuration Agent

Configuring the Cisco Networking Services Event and EXEC Agents

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns config partial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**source** *interface name*] [**inventory**]
4. **logging cns-events** [*severity-level*]
5. **cns exec** [**encrypt**] [*port-number*] [**source** {*ip-address* | *interface-type-number*}]
6. **cns event** {*hostname* | *ip-address*} [**encrypt**] [*port-number*] [**backup**] [**failover-time** *seconds*] [**keepalive** *seconds retry-count*] [**source** *ip-address* | *interface-name*][**clock-timeout** *time*] [**reconnect-time** *time*]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cns config partial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**source** *interface name*] [**inventory**] **Example:** `Device(config)# cns config partial 172.28.129.22 80`	(Optional) Starts the Cisco Networking Services configuration agent, which provides Cisco Networking Services configuration services to Cisco clients, and initiates an incremental (partial) configuration. • Use the optional *port-number* argument to specify the port number for the configuration server. The default is 80. • Use the optional **source** keyword and *ip-address* argument to specify the use of an IP address as the source for Cisco Networking Services configuration agent communications. • Use the optional **inventory** keyword to send an inventory of the linecards and modules in the device to the Cisco Networking Services configuration engine as part of the HTTP request. **Note** The optional **encrypt** keyword is available only in images that support SSL.
Step 4	**logging cns-events** [*severity-level*] **Example:** `Device(config)# logging cns-events 2`	(Optional) Enables XML-formatted system event message logging to be sent through the Cisco Networking Services event bus. • Use the optional *severity-level* argument to specify the number or name of the desired severity level at which messages should be logged. The default is level 7 (debugging).
Step 5	**cns exec** [**encrypt**] [*port-number*] [**source** {*ip-address* | *interface-type-number*}] **Example:** `Device(config)# cns exec source 172.17.2.2`	(Optional) Enables and configures the Cisco Networking Services EXEC agent, which provides Cisco Networking Services EXEC services to Cisco clients. • Use the optional *port-number* argument to specify the port number for the EXEC server. The default is 80. • Use the optional **source** keyword and *ip-address*/*interface-type number* argument to specify the use of an IP address as the source for Cisco Networking Services EXEC agent communications.

	Command or Action	Purpose
		Note The optional **encrypt keyword** is available only in images that support SSL.
Step 6	**cns event** {*hostname* | *ip-address*} [**encrypt**] [*port-number*] [**backup**] [**failover-time** *seconds*] [**keepalive** *seconds retry-count*] [**source** *ip-address* | *interface-name*][**clock-timeout** *time*] [**reconnect-time** *time*] **Example:** `Device(config)# cns event 172.28.129.22 source 172.22.2.1`	Configures the Cisco Networking Services event gateway, which provides Cisco Networking Services event services to Cisco clients. • The optional **encrypt** keyword is available only in images that support SSL. • Use the optional *port-number* argument to specify the port number for the event server. The default is 11011 with no encryption and 11012 with encryption. • Use the optional **backup** keyword to indicate that this is the backup gateway. Before configuring a backup gateway, ensure that a primary gateway is configured. • Use the optional **failover-time** keyword and *seconds* argument to specify a time interval in seconds to wait for the primary gateway route after the route to the backup gateway is established. • Use the optional **keepalive** keyword with the *seconds* and *retry-count* arguments to specify the keepalive timeout in seconds and the retry count. • Use the optional **source** keyword and *ip-address*/*interface-name* argument to specify the use of an IP address as the source for Cisco Networking Services event agent communications. • Use the optional **clock-timeout** keyword to specify the maximum time, in minutes, that the Cisco Networking Services event agent will wait for the clock to be set for transports (such as SSL) that require an accurate clock. • Use the optional **reconnect-time** keyword to specify the configurable upper limit of the maximum retry timeout. **Note** Until the **cns event** command is entered, no transport connections to the Cisco Networking Services event bus are made and therefore no other Cisco Networking Services agents are operational.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

- Use the **show cns event connections** command to check that the Cisco Networking Services event agent is connected to the Cisco Networking Services event gateway.
- Use the **show cns event subject** command to check that the image agent subject names are registered. Subject names for the Cisco Networking Services image agent begin with cisco.mgmt.cns.image.

Configuration Examples for CNS Configuration Agent

Example: Enabling and Configuring Cisco Networking Services Agents

The following example shows various Cisco Networking Services agents being enabled and configured starting with the configuration agent being enabled with the **cns config partial** command to configure an incremental (partial) configuration on a remote device. The Cisco Networking Services configuration engine IP address is 172.28.129.22, and the port number is 80. The Cisco Networking Services exec agent is enabled with an IP address of 172.28.129.23, and the Cisco Networking Services event agent is enabled with an IP address of 172.28.129.24. Until the Cisco Networking Services event agent is enabled, no other Cisco Networking Services agents are operational.

```
cns config partial 172.28.129.22 80
cns exec 172.28.129.23 source 172.22.2.2
cns event 172.28.129.24 source 172.22.2.1
exit
```

In the following example, the Cisco Networking Services image agent parameters are configured using the CLI. An image ID is specified to use the IP address of the GigabitEthernet interface 0/1/1, a password is configured for the Cisco Networking Services image agent services, the Cisco Networking Services image upgrade retry interval is set to four minutes, and image management and status servers are configured.

```
cns id GigabitEthernet0/1/1 ipaddress image
cns image retry 240
cns image password abctext
cns image server https://10.21.2.3/cns/imgsvr status https://10.21.2.3/cns/status/
```

In the following example, the Cisco Networking Services image agent is configured to use the Cisco Networking Services Event Bus. An image ID is specified as the hardware serial number of the networking device, the Cisco Networking Services event agent is enabled with a number of parameters, and the Cisco Networking Services image agent is enabled without any keywords or options. The Cisco Networking Services image agent will listen for events on the Cisco Networking Services Event Bus.

```
cns id hardware-serial image
cns event 10.21.9.7 11011 keepalive 240 120 failover-time 5
cns image
cns image password abctext
```

Example: Retrieving a Cisco Networking Services Image from a Server

In the following example, the Cisco Networking Services image agent polls a file server using the **cns image retrieve** command. Assuming that the Cisco Networking Services image agent is already enabled, the file server and status server paths specified here will overwrite any existing image agent server and status configuration. The new file server will be polled and a new image, if it exists, will be downloaded to the networking device.

```
cns image retrieve server https://10.19.2.3/cns/ status https://10.19.2.3/cnsstatus/
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Standards and RFCs

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

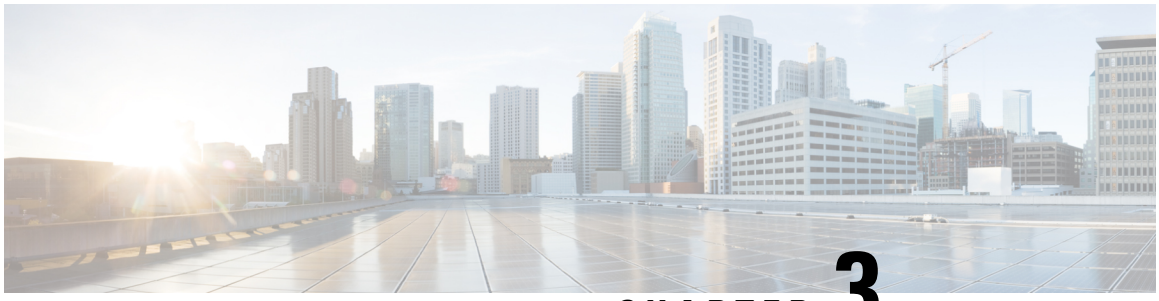
Feature Information for CNS Configuration Agent

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3: Feature Information for CNS Configuration Agent

Feature Name	Releases	Feature Information
CNS Configuration Agent	Cisco IOS XE Release 2.1 12.0(18)ST 12.0(22)S 12.2(2)T 12.2(8)T 12.2(33)SRA 12.2(33)SB 12.2(33)SXI	The Cisco Networking Services Configuration Agent feature supports routing devices by providing the following: • Initial configurations • Incremental (partial) configurations • Synchronized configuration updates The following commands were introduced or modified by this feature: **cns config cancel**, **cns config initial** , **cns config partial** , **cns config retrieve** ,**cns password**, **debug cns config**, **debug cns xml-parser** , **show cns config outstanding** ,**show cns config stats**, **show cns config status** .



CHAPTER 3

Cisco Networking Services Config Retrieve Enhancement with Retry and Interval

Information About CNS Config Retrieve Enhancement with Retry and Interval

Cisco Networking Services Config Retrieve Enhancement with Retry and Interval

The Cisco Networking Services Config Retrieve Enhancement with Retry and Interval feature adds new functionality to the **cns config retrieve** command enabling you to specify the retry interval and an amount of time in seconds to wait before attempting to retrieve a configuration from a trusted server.

How to Configure CNS Config Retrieve Enhancement with Retry and Interval

Retrieving a Cisco Networking Services Configuration from a Server

Use this task to request the configuration of a device from a configuration server. Use the **cns trusted-server** command to specify which configuration server can be used (trusted).

Before you begin

This task assumes that you have specified a trusted server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns config retrieve** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**overwrite-startup**] [**retry** *retries* **interval** *seconds*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cns config retrieve** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**overwrite-startup**] [**retry** *retries* **interval** *seconds*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**] **Example:** `Device(config)# cns config retrieve server1 retry 5 interval 45`	Allows the device to retrieve configuration data from a web server. • The **retry** keyword is a number in the range 1 to 100, and will prompt for an **interval** in the range 1 to 3600 seconds. **Note** Troubleshooting Tips If you need to stop the retrieval process, enter the Ctrl+Shift+6 key sequence.

Configuration Examples for CNS Config Retrieve Enhancement with Retry and Interval

Example: Retrieving a Cisco Networking Services Configuration from a Server

Retrieving Configuration Data from the Cisco Networking Services Trusted Server

The following example shows how to request a configuration from a trusted server at 10.1.1.1:

```
cns trusted-server all 10.1.1.1
exit
cns config retrieve 10.1.1.1
```

The following example shows how to request a configuration from a trusted server at 10.1.1.1 and to configure a Cisco Networking Services configuration retrieve interval using the **cns config retrieve** command:

```
cns trusted-server all 10.1.1.1
exit
cns config retrieve 10.1.1.1 retry 50 interval 1500
CNS Config Retrieve Attempt 1 out of 50 is in progress
Next cns config retrieve retry is in 1499 seconds (Ctrl-Shft-6 to abort this command).
..
00:26:40: %CNS-3-TRANSPORT: CNS_HTTP_CONNECTION_FAILED:10.1.1.1 -Process= "CNS config retv",
 ipl= 0, pid= 43
```

00:26:40: %CNS-3-TRANSPORT: CNS_HTTP_CONNECTION_FAILED -Process= "CNS config retv", ipl= 0, pid= 43.....

```
cns config retrieve 10.1.1.1
```

Applying the Retrieved Data to the Running Configuration File

The following example shows how to check and apply configuration data retrieved from the server to running configuration file only. The Cisco Networking Services Configuration Agent will attempt to retrieve configuration data at 30-second intervals until the attempt is successful, or is unsuccessful five times in these attempts.

```
cns config retrieve 10.1.1.1 syntax-check no-persist retry 5 interval 30
```

Overwriting the Startup Configuration File with the Retrieved Data

The following example shows how to overwrite the startup configuration file with the configuration data retrieved from the server. The configuration data will not be applied to the running configuration.

```
cns config retrieve 10.1.1.1 syntax-check no-persist retry 5 interval 30
cns config retrieve 10.1.1.1 overwrite-startup
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Standards and RFCs

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

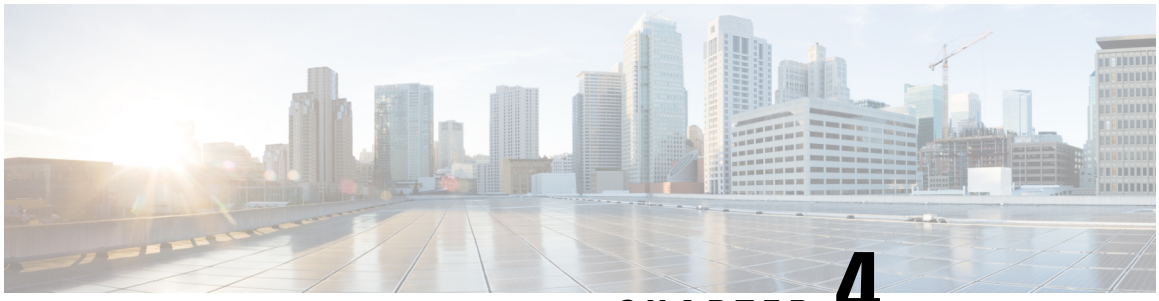
Feature Information for CNS Config Retrieve Enhancement with Retry and Interval

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for Cisco Networking Services Config Retrieve Enhancement with Retry and Interval

Feature Name	Releases	Feature Information
Cisco Networking Services Config Retrieve Enhancement with Retry and Interval	Cisco IOS XE Release 2.1 12.4(15)T 12.2(33)SRC 12.2(33)SB 12.2(50)SY	The Cisco Networking Services Config Retrieve Enhancement with Retry and Interval feature adds two options to the **cns config retrieve** command enabling you to specify an amount of time in seconds to wait before attempting to retrieve a configuration from a trusted server. The number of retries is restricted to 100 to prevent the configuration agent from indefinitely attempting to reach an unreachable server. Use the keyboard combination **Ctrl-Shift-6** to abort the **cns config retrieve** command. The following command was modified by this feature: **cns config retrieve**.



CHAPTER 4

Cisco Networking Services Interactive CLI

Information About CNS Interactive CLI

Cisco Networking Services Interactive CLI

The Cisco Networking Services Interactive CLI feature provides a XML interface that allows you to send interactive commands to a device, such as commands that generate prompts for user input. A benefit of this feature is that interactive commands can be aborted before they have been fully processed. For example, for commands that generate a significant amount of output, the XML interface can be customized to limit the size of the output or the length of time allowed for the output to accumulate. The capability to use a programmable interface to abort a command before its normal termination (similar to manually aborting a command) can greatly increase the efficiency of diagnostic applications that might use this functionality. The new XML interface also allows for multiple commands to be processed in a single session. The response for each command is packaged together and sent in a single response event.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

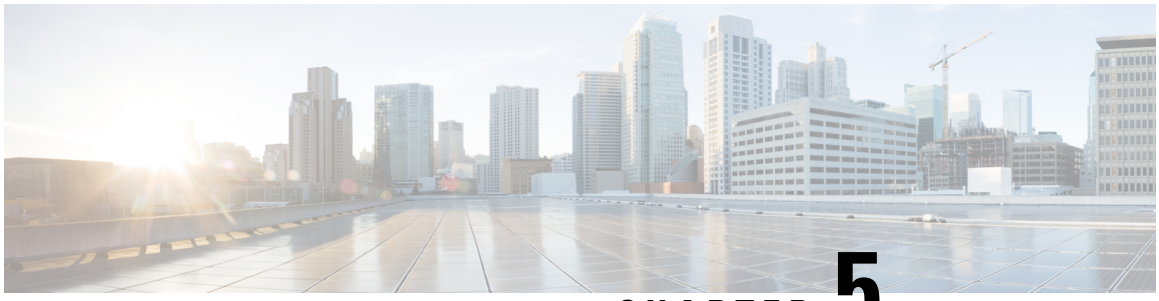
Feature Information for CNS Interactive CLI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for Cisco Networking Services Interactive CLI

Feature Name	Releases	Feature Information
Cisco Networking Services Interactive CLI	Cisco IOS XE Release 2.1 12.0(28)S 12.2(18)SXE 12.2(18)SXF2 12.2(33)SRC 12.2(33)SXI	The Cisco Networking Services Interactive CLI feature introduces an XML interface that allows you to send interactive commands to a device, such as commands that generate prompts for user input.



CHAPTER 5

Command Scheduler (Kron)

Restrictions for Command Scheduler

The EXEC CLI specified in a Command Scheduler policy list must neither generate a prompt nor can it be terminated using keystrokes. Command Scheduler is designed as a fully automated facility, and no manual intervention is permitted.

Information About Command Scheduler (Kron)

Command Scheduler

The Command Scheduler (KRON) Policy for System Startup feature enables support for the Command Scheduler upon system startup.

The Command Scheduler allows customers to schedule fully-qualified EXEC mode CLI commands to run once, at specified intervals, at specified calendar dates and times, or upon system startup. Originally designed to work with Cisco Networking Services commands, Command Scheduler now has a broader application. Using the Cisco Networking Services image agent feature, remote devices residing outside a firewall or using Network Address Translation (NAT) addresses can use Command Scheduler to launch CLI at intervals, to update the image running in the device.

Command Scheduler has two basic processes. A policy list is configured containing lines of fully-qualified EXEC CLI commands to be run at the same time or same interval. One or more policy lists are then scheduled to run after a specified interval of time, at a specified calendar date and time, or upon system startup. Each scheduled occurrence can be set to run either once only or on a recurring basis.

How to Configure Command Scheduler (Kron)

Configuring Command Scheduler Policy Lists and Occurrences

An occurrence for Command Scheduler is defined as a scheduled event. Policy lists are configured to run after a specified interval of time, at a specified calendar date and time, or upon system startup. Policy lists can be run once, as a one-time event, or as recurring events over time.

Command Scheduler occurrences can be scheduled before the associated policy list has been configured, but a warning will advise you to configure the policy list before it is scheduled to run.

Before you begin

Perform this task to set up Command Scheduler policy lists of EXEC Cisco Networking Services commands and configure a Command Scheduler occurrence to specify the time or interval after which the Cisco Networking Services commands will run.

Command Scheduler Policy Lists

Policy lists consist of one or more lines of fully-qualified EXEC CLI commands. All commands in a policy list are executed when the policy list is run by Command Scheduler using the **kron occurrence** command. Use separate policy lists for CLI commands that are run at different times. No editor function is available, and the policy list is run in the order in which it was configured. To delete an entry, use the **no** form of the **cli** command followed by the appropriate EXEC command. If an existing policy list name is used, new entries are added to the end of the policy list. To view entries in a policy list, use the **show running-config** command. If a policy list is scheduled to run only once, it will not be displayed by the **show running-config** command after it has run.

Policy lists can be configured after the policy list has been scheduled, but each policy list must be configured before it is scheduled to run.

Command Scheduler Occurences

The clock time must be set on the routing device before a Command Scheduler occurrence is scheduled to run. If the clock time is not set, a warning message will appear on the console screen after the **kron occurrence** command has been entered. Use the **clock** command or Network Time Protocol (NTP) to set the clock time.

The EXEC CLI to be run by Command Scheduler must be tested on the routing device to determine if it will run without generating a prompt or allowing execution interruption by keystrokes. Initial testing is important because Command Scheduler will delete the entire policy list if any CLI syntax fails. Removing the policy list ensures that any CLI dependencies will not generate more errors.

If you use the **conditional** keyword with the **kron policy-list** command, execution of the commands will stop when an error is encountered.

Note

- No more than 31 policy lists can be scheduled to run at the same time.
- If a one-time occurrence is scheduled, the occurrence will not be displayed by the **show running-config** command after the occurrence has run.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **kron policy-list** *list-name* [**conditional**]
4. **cli** *command*
5. **exit**
6. **kron occurrence** *occurrence-name* [**user** *username*] {**in**[[*numdays:*]*numhours:*]*nummin*| **at** *hours:min*[[*month*] *day-of-month*] [*day-of-week*]} {**oneshot**| **recurring**| **system-startup**}
7. **policy-list** *list-name*
8. **exit**
9. **show kron schedule**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**kron policy-list** *list-name* [**conditional**] **Example:** `Device(config)# kron policy-list cns-weekly`	Specifies a name for a new or existing Command Scheduler policy list and enters kron-policy configuration mode. • If the *list-name* is new, a new policy list structure is created. • If the *list-name* exists, the existing policy list structure is accessed. The policy list is run in configured order with no editor function. • If the optional **conditional** keyword is used, execution of the commands stops when an error is encountered.
Step 4	**cli** *command* **Example:** `Device(config-kron-policy)# cli cns image retrieve server https://10.19.2.3/cnsweek/ status https://10.19.2.3/cnsstatus/week/`	Specifies the fully-qualified EXEC command and associated syntax to be added as an entry in the specified Command Scheduler policy list. • Each entry is added to the policy list in the order in which it is configured. • Repeat this step to add other EXEC CLI commands to a policy list to be executed at the same time or interval.

	Command or Action	Purpose
		Note EXEC commands that generate a prompt or can be terminated using keystrokes will cause an error.
Step 5	**exit** **Example:** `Device(config-kron-policy)# exit`	Exits kron-policy configuration mode and returns the device to global configuration mode.
Step 6	**kron occurrence** *occurrence-name* [**user** *username*] {**in**[[*numdays:*]*numhours:*]*nummin*| **at** *hours:min*[[*month*] *day-of-month*] [*day-of-week*]} {**oneshot**| **recurring**| **system-startup**} **Example:** `Device(config)# kron occurrence may user sales at 6:30 may 20 oneshot`	Specifies a name and schedule for a new or existing Command Scheduler occurrence and enters kron-occurrence configuration mode. • Use the **in** keyword to specify a delta time interval with a timer that starts when this command is configured. • Use the **at** keyword to specify a calendar date and time. • Choose either the **oneshot** or **recurring** keyword to schedule Command Scheduler occurrence once or repeatedly. Add the optional **system-startup** keyword for the occurrence to be at system startup.
Step 7	**policy-list** *list-name* **Example:** `Device(config-kron-occurrence)# policy-list sales-may`	Specifies a Command Scheduler policy list. • Each entry is added to the occurrence list in the order in which it is configured. **Note** If the CLI commands in a policy list generate a prompt or can be terminated using keystrokes, an error will be generated and the policy list will be deleted.
Step 8	**exit** **Example:** `Device(config-kron-occurrence)# exit`	Exits kron-occurrence configuration mode and returns the device to global configuration mode. • Repeat this step to exit global configuration mode.
Step 9	**show kron schedule** **Example:** `Device# show kron schedule`	(Optional) Displays the status and schedule information of Command Scheduler occurrences.

Examples

In the following example, output information is displayed about the status and schedule of all configured Command Scheduler occurrences:

```
Device# show kron schedule
Kron Occurrence Schedule
cns-weekly inactive, will run again in 7 days 01:02:33
may inactive, will run once in 32 days 20:43:31 at 6:30 on May 20
```

Troubleshooting Tips

Use the **debug kron** command in privileged EXEC mode to troubleshoot Command Scheduler command operations. Use any debugging command with caution because the volume of output generated can slow or stop the device's operations.

Configuration Examples for Command Scheduler (Kron)

Example: Command Scheduler Policy Lists and Occurrences

In the following example, a Command Scheduler policy named cns-weekly is configured to run two sets of EXEC CLI involving Cisco Networking Services commands. The policy is then scheduled with two other policies to run every seven days, one hour and thirty minutes.

```
kron policy-list cns-weekly
cli cns image retrieve server http://10.19.2.3/week/ status http://10.19.2.5/status/week/
cli cns config retrieve page /testconfig/config.asp no-persist
exit
kron occurrence week in 7:1:30 recurring
policy-list cns-weekly
policy-list itd-weekly
policy-list mkt-weekly
```

In the following example, a Command Scheduler policy named sales-may is configured to run a Cisco Networking Services command to retrieve a specified image from a remote server. The policy is then scheduled to run only once on May 20, at 6:30 a.m.

```
kron policy-list sales-may
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence may at 6:30 May 20 oneshot
policy-list sales-may
```

In the following example, a Command Scheduler policy named image-sunday is configured to run a Cisco Networking Services command to retrieve a specified image from a remote server. The policy is then scheduled to run every Sunday at 7:30 a.m.

```
kron policy-list image-sunday
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence sunday user sales at 7:30 sunday recurring
policy-list image-sunday
```

In the following example, a Command Scheduler policy named file-retrieval is configured to run a Cisco Networking Services command to retrieve a specific file from a remote server. The policy is then scheduled to run on system startup.

```
kron policy-list file-retrieval
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence system-startup
policy-list file-retrieval
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Standards and RFCs

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

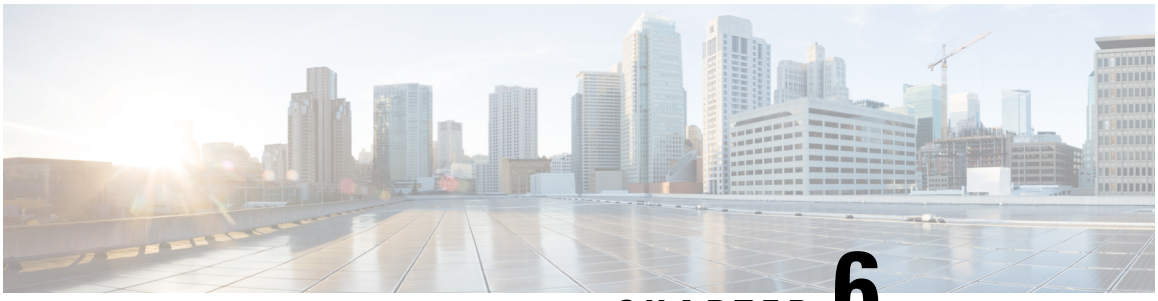
Feature Information for Command Scheduler (Kron)

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for Command Scheduler (Kron)

Feature Name	Releases	Feature Information
Command Scheduler (Kron)	Cisco IOS XE Release 2.1 12.3(1) 12.2(33)SRA 12.2(33)SRC 12.2(33)SB 12.2(33)SXI 12.2(50)SY	The Command Scheduler feature provides the ability to schedule some EXEC CLI commands to run at specific times or at specified intervals. The following commands were introduced or modified by this feature: **cli**, **debug kron**, **kron occurrence**, **kron policy-list**, **policy-list**, **show kron schedule**.
Command Scheduler (Kron) Policy for System Startup	12.2(33)SRC 12.2(50)SY 12.2(33)SB 12.4(15)T	The Command Scheduler (Kron) Policy for System Startup feature enables support for the Command Scheduler feature upon system startup.



CHAPTER 6

Network Configuration Protocol

The Network Configuration Protocol (NETCONF) defines a simple mechanism through which a network device can be managed, configuration data can be retrieved, and new configuration data can be uploaded and manipulated. NETCONF uses Extensible Markup Language (XML)-based data encoding for the configuration data and protocol messages.

Prerequisites for NETCONF

A vty line must be available for each NETCONF session as specified by the **netconf max-session** command.

Information About NETCONF

NETCONF Notifications

NETCONF sends notifications of any configuration change over NETCONF. A notification is an event indicating that a configuration change has occurred. The change can be a new configuration, deleted configuration, or changed configuration. The notifications are sent at the end of a successful configuration operation as one message that shows the set of changes rather than showing individual messages for each line that is changed in the configuration.

How to Configure NETCONF

Configuring the NETCONF Network Manager Application

Step 1 Use the following CLI string to configure the NETCONF network manager application to invoke NETCONF as an SSH subsystem:

Example:

```
Unix Side: ssh-2 -s companyname@10.1.1.1 netconf
```

Step 2 As soon as the NETCONF session is established, indicate the server capabilities by sending an XML document containing a <hello>:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
    <hello>
      <capabilities>
        <capability>
            urn:ietf:params:xml:ns:netconf:base:1.0
          </capability>
          <capability>
            urn:ietf:params:ns:netconf:capability:startup:1.0
          </capability>
       </capabilities>
    <session-id>4<session-id>
</hello>]]>]]>
```

The client also responds by sending an XML document containing a <hello>:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
 <hello>
   <capabilities>
       <capability>
           urn:ietf:params:xml:ns:netconf:base:1.0
     </capability>
    </capabilities>
</hello>]]>]]>
```

Note Although the example shows the server sending a <hello> message followed by the message from the client, both sides send the message as soon as the NETCONF subsystem is initialized, perhaps simultaneously.

Tip All NETCONF requests must end with]]>]]> which denotes an end to the request. Until the]]>]]> sequence is sent, the device will not process the request.

See the “Example: Configuring NETCONF over SSHv2” section for a specific example.

Step 3 Use the following XML string to enable the NETCONF network manager application to send and receive NETCONF notifications:

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.0"><notification-on/>
</rpc>]]>]]>
```

Step 4 Use the following XML string to stop the NETCONF network manager application from sending or receiving NETCONF notifications:

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.13"><notification-off/>
</rpc>]]>]]>
```

Delivering NETCONF Payloads

Use the following XML string to deliver the NETCONF payload to the network manager application:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.cisco.com/cpi_10/schema" elementFormDefault="qualified"
 attributeFormDefault="unqualified" xmlns="http://www.cisco.com/cpi_10/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <!--The following elements define the cisco extensions for the content of the filter
element in a <get-config> request. They allow the client to specify the format of the
response and to select subsets of the entire configuration to be included.-->
   <xs:element name="config-format-text-block">
      <xs:annotation>
         <xs:documentation>If this element appears in the filter, then the client is
requesting that the response data be sent in config command block format.</xs:documentation>

      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="text-filter-spec" minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="config-format-text-cmd">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="text-filter-spec"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="config-format-xml">
      <xs:annotation>
        <xs:documentation>When this element appears in the filter of a get-config request,
 the results are to be returned in E-DI XML format. The content of this element is treated
 as a filter.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:complexContent>
            <xs:extension base="xs:anyType"/>
         </xs:complexContent>
      </xs:complexType>
   </xs:element>
   <!--These elements are used in the filter of a <get> to specify operational data to
return.-->
   <xs:element name="oper-data-format-text-block">
```

```
      <xs:complexType>
         <xs:sequence>
            <xs:element name="show" type="xs:string" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="oper-data-format-xml">
      <xs:complexType>
         <xs:sequence>
            <xs:any/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <!--When confing-format-text format is specified, the following describes the content
of the data element in the response-->
   <xs:element name="cli-config-data">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="cmd" type="xs:string" maxOccurs="unbounded">
               <xs:annotation>
                  <xs:documentation>Content is a command. May be multiple
lines.</xs:documentation>
               </xs:annotation>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="cli-config-data-block" type="xs:string">
      <xs:annotation>
         <xs:documentation>The content of this element is the device configuration as it
would be sent to a terminal session. It contains embedded newline characters that must be
preserved as they represent the boundaries between the individual command
lines</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="text-filter-spec">
      <xs:annotation>
         <xs:documentation>If this element is included in the config-format-text element,
then the content is treated as if the string was appended to the "show running-config"
command line.</xs:documentation>
      </xs:annotation>
   </xs:element>
   <xs:element name="cli-oper-data-block">
      <xs:complexType>
         <xs:annotation>
            <xs:documentation> This element is included in the response to get operation.
Content of this element is the operational data in text format.</xs:documentation>
         </xs:annotation>
         <xs:sequence>
            <xs:element name="item" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="exec"/>
                     <xs:element name="show"/>
                     <xs:element name="response"/>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:schema>
```

Formatting NETCONF Notifications

The NETCONF network manager application uses .xsd schema files to describe the format of the XML NETCONF notification messages that are sent between a NETCONF network manager application and a device running NETCONF over SSHv2 or BEEP. These files can be displayed in a browser or a schema reading tool. You can use these schemas to validate that the XML is correct. These schemas describe the format, not the content, of the data being exchanged.

NETCONF uses the <edit-config> function to load all of a specified configuration to a specified target configuration. When this new configuration is entered, the target configuration is not replaced. The target configuration is changed according to the data and requested operations of the requesting source.

The following are schemas for the NETCONF <edit-config> function in CLI, CLI block, and XML format.

NETCONF <edit-config> Request: CLI Format

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cli-config-data>
<cmd>hostname test</cmd>
            <cmd>interface fastEthernet0/1</cmd>
            <cmd>ip address 192.168.1.1 255.255.255.0</cmd>
</cli-config-data>
      </config>
   </edit-config>
</rpc>]]>]]>
```

NETCONF <edit-config> Response: CLI Format

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:netconf:base:1.0">
   <ok/>
</rpc-reply>]]>]]>
```

NETCONF <edit-config> Request: CLI-Block Format

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="netconf.mini.edit.3">
   <edit-config>
      <target>
         <running/>
      </target>
      <config>
         <cli-config-data-block>
            hostname bob
            interface fastEthernet0/1
            ip address 192.168.1.1 255.255.255.0
         </cli-config-data-block>
      </config>
   </edit-config>
</rpc>]]>]]>
```

NETCONF <edit-config> Response: CLI-Block Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="netconf.mini.edit.3" xmlns="urn:ietf:params:netconf:base:1.0">
   <ok/>
</rpc-reply>]]>]]>
```

The following are schemas for the NETCONF <get-config> function in CLI and CLI-block format.

NETCONF <get-config> Request: CLI Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get-config>
      <source>
         <running/>
      </source>
      <filter>
         <config-format-text-cmd>
            <text-filter-spec> | inc interface </text-filter-spec>
         </config-format-text-cmd>
</filter>
   </get-config>
</rpc>]]>]]>
```

NETCONF <get-config> Response: CLI Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <data>
      <cli-config-data>
         <cmd>interface FastEthernet0/1</cmd>
         <cmd>interface FastEthernet0/2</cmd>
      </cli-config-data>
   </data>
</rpc-reply>]]>]]>
```

NETCONF <get-config> Request: CLI-Block Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get-config>
      <source>
         <running/>
      </source>
      <filter>
         <config-format-text-block>
            <text-filter-spec> | inc interface </text-filter-spec>
         </config-format-text-block>
      </filter>
   </get-config>
</rpc>]]>]]>
```

NETCONF <get-config> Response: CLI-Block Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <data>
```

```
      <cli-config-data-block>
         interface FastEthernet0/1
         interface FastEthernet0/2
      </cli-config-data-block>
   </data>
</rpc-reply>]]>]]>
```

NETCONF uses the <get> function to retrieve configuration and device-state information. The NETCONF <get> format is the equivalent of a Cisco IOS **show** command. The <filter> parameter specifies the portion of the system configuration and device-state data to retrieve. If the <filter> parameter is empty, nothing is returned.

The following are schemas for the <get> function in CLI and CLI-block format.

NETCONF <get> Request: CLI Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get>
       <filter>
          <config-format-text-cmd>
             <text-filter-spec> | include interface </text-filter-spec>
          </config-format-text-cmd>
          <oper-data-format-text-block>
             <exec>show interfaces</exec>
             <exec>show arp</exec>
          </oper-data-format-text-block>
       </filter>
    </get>
 </rpc>]]>]]>
```

NETCONF <get> Response: CLI Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
       <cli-config-data>
<cmd>interface Loopback0</cmd>
<cmd>interface GigabitEthernet0/1</cmd>
<cmd>interface GigabitEthernet0/2</cmd>
</cli-config-data>
<cli-oper-data-block>
          <item>
             <exec>show interfaces</exec>
             <response>
                <!-- output of "show interfaces" ------>
             </response>
          </item>
          <item>
             <exec>show arp</exec>
             <response>
                <!-- output of "show arp" ------>
             </response>
          </item>
       </cli-oper-data-block>
    </data>
</rpc-reply>]]>]]>
```

NETCONF <get> Request: CLI-Block Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <get>
       <filter>
          <config-format-text-block>
             <text-filter-spec> | include interface </text-filter-spec>
          </config-format-text-block>
          <oper-data-format-text-block>
             <exec>show interfaces</exec>
             <exec>show arp</exec>
          </oper-data-format-text-block>
       </filter>
    </get>
 </rpc>]]>]]>
```

NETCONF <get> Response: CLI-Block Format

```
<?xml version="1.0" encoding=\"UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <data>
       <cli-config-data-block>
interface Loopback0
interface GigabitEthernet0/1
interface GigabitEthernet0/2
       </cli-config-data-block>
       <cli-oper-data-block>
          <item>
             <exec>show interfaces</exec>
             <response>
                <!-- output of "show interfaces" ------>
             </response>
          </item>
          <item>
             <exec>show arp</exec>
             <response>
                <!-- output of "show arp" ------>
             </response>
          </item>
       </cli-oper-data-block>
    </data>
</rpc-reply>]]>]]>
```

Monitoring and Maintaining NETCONF Sessions

Note

- A minimum of four concurrent NETCONF sessions must be configured.
- A maximum of 16 concurrent NETCONF sessions can be configured.
- NETCONF does not support SSHv1.

SUMMARY STEPS

1. **enable**
2. **show netconf** {**counters** | **session**| **schema**}

3. **debug netconf** {**all** | **error**}
4. **clear netconf** {**counters** | **sessions**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show netconf** {**counters**	**session**| **schema**} **Example:** `Device# show netconf counters`	Displays NETCONF information.
Step 3	**debug netconf** {**all**	**error**} **Example:** `Device# debug netconf error`	Enables debugging of NETCONF sessions.
Step 4	**clear netconf** {**counters**	**sessions**} **Example:** `Device# clear netconf sessions`	Clears NETCONF statistics counters and NETCONF sessions, and frees associated resources and locks.

Configuration Examples for NETCONF

Example: Configuring the NETCONF Network Manager Application

The following example shows how to configure the NETCONF network manager application to invoke NETCONF as an SSH subsystem:

```
Unix Side: ssh-2 -s companyname@10.1.1.1 netconf
```

As soon as the NETCONF session is established, indicate the server capabilities by sending an XML document containing a <hello>:

```
<?xml version="1.0" encoding="UTF-8"?>
    <hello>
      <capabilities>
        <capability>
            urn:ietf:params:xml:ns:netconf:base:1.0
          </capability>
          <capability>
            urn:ietf:params:ns:netconf:capability:startup:1.0
          </capability>
       </capabilities>
```

```
    <session-id>4<session-id>
</hello>]]>]]>
```

The client also responds by sending an XML document containing a <hello>:

```
<?xml version="1.0" encoding="UTF-8"?>
 <hello>
   <capabilities>
       <capability>
           urn:ietf:params:xml:ns:netconf:base:1.0
     </capability>
    </capabilities>
</hello>]]>]]>
```

Use the following XML string to enable the NETCONF network manager application to send and receive NETCONF notifications:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.0"><notification-on/>
</rpc>]]>]]>
```

Use the following XML string to stop the NETCONF network manager application from sending or receiving NETCONF notifications:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.13"><notification-off/>
</rpc>]]>]]>
```

Example: Monitoring NETCONF Sessions

The following is sample output from the **show netconf counters** command:

```
Device# show netconf counters
NETCONF Counters
Connection Attempts:0: rejected:0 no-hello:0 success:0
Transactions
        total:0, success:0, errors:0
detailed errors:
        in-use 0        invalid-value 0         too-big 0
        missing-attribute 0     bad-attribute 0         unknown-attribute 0
        missing-element 0       bad-element 0   unknown-element 0
        unknown-namespace 0     access-denied 0         lock-denied 0
        resource-denied 0       rollback-failed 0       data-exists 0
        data-missing 0  operation-not-supported 0       operation-failed 0
        partial-operation 0
```

The following is sample output from the **show netconf session** command:

```
Device# show netconf session
(Current | max) sessions:   3 | 4
Operations received: 100              Operation errors: 99
Connection Requests: 5                Authentication errors: 2   Connection Failures: 0
ACL dropped : 30
Notifications  Sent: 20
```

The output of the **show netconf schema** command displays the element structure for a NETCONF request and the resulting reply. This schema can be used to construct proper NETCONF requests and parse the resulting replies. The nodes in the schema are defined in RFC 4741. The following is sample output from the **show netconf schema** command:

```
Device# show netconf schema
New Name Space 'urn:ietf:params:xml:ns:netconf:base:1.0'
<VirtualRootTag> [0, 1] required
  <rpc-reply> [0, 1] required
    <ok> [0, 1] required
    <data> [0, 1] required
    <rpc-error> [0, 1] required
      <error-type> [0, 1] required
      <error-tag> [0, 1] required
      <error-severity> [0, 1] required
      <error-app-tag> [0, 1] required
      <error-path> [0, 1] required
      <error-message> [0, 1] required
      <error-info> [0, 1] required
        <bad-attribute> [0, 1] required
        <bad-element> [0, 1] required
        <ok-element> [0, 1] required
        <err-element> [0, 1] required
        <noop-element> [0, 1] required
        <bad-namespace> [0, 1] required
        <session-id> [0, 1] required
  <hello> [0, 1] required
    <capabilities> 1 required
      <capability> 1+ required
  <rpc> [0, 1] required
    <close-session> [0, 1] required
    <commit> [0, 1] required
      <confirmed> [0, 1] required
      <confirm-timeout> [0, 1] required
    <copy-config> [0, 1] required
      <source> 1 required
        <config> [0, 1] required
          <cli-config-data> [0, 1] required
            <cmd> 1+ required
          <cli-config-data-block> [0, 1] required
          <xml-config-data> [0, 1] required
            <Device-Configuration> [0, 1] required
              <> any subtree is allowed
        <candidate> [0, 1] required
        <running> [0, 1] required
        <startup> [0, 1] required
        <url> [0, 1] required
      <target> 1 required
        <candidate> [0, 1] required
        <running> [0, 1] required
        <startup> [0, 1] required
        <url> [0, 1] required
    <delete-config> [0, 1] required
      <target> 1 required
        <candidate> [0, 1] required
        <running> [0, 1] required
        <startup> [0, 1] required
        <url> [0, 1] required
    <discard-changes> [0, 1] required
    <edit-config> [0, 1] required
      <target> 1 required
        <candidate> [0, 1] required
        <running> [0, 1] required
        <startup> [0, 1] required
        <url> [0, 1] required
      <default-operation> [0, 1] required
      <test-option> [0, 1] required
      <error-option> [0, 1] required
```

```
  <config> 1 required
    <cli-config-data> [0, 1] required
      <cmd> 1+ required
    <cli-config-data-block> [0, 1] required
    <xml-config-data> [0, 1] required
      <Device-Configuration> [0, 1] required
        <> any subtree is allowed
<get> [0, 1] required
  <filter> [0, 1] required
    <config-format-text-cmd> [0, 1] required
      <text-filter-spec> [0, 1] required
    <config-format-text-block> [0, 1] required
      <text-filter-spec> [0, 1] required
    <config-format-xml> [0, 1] required
    <oper-data-format-text-block> [0, 1] required
      <exec> [0, 1] required
      <show> [0, 1] required
    <oper-data-format-xml> [0, 1] required
      <exec> [0, 1] required
      <show> [0, 1] required
<get-config> [0, 1] required
  <source> 1 required
    <config> [0, 1] required
      <cli-config-data> [0, 1] required
        <cmd> 1+ required
      <cli-config-data-block> [0, 1] required
      <xml-config-data> [0, 1] required
        <Device-Configuration> [0, 1] required
          <> any subtree is allowed
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
  <filter> [0, 1] required
    <config-format-text-cmd> [0, 1] required
      <text-filter-spec> [0, 1] required
    <config-format-text-block> [0, 1] required
      <text-filter-spec> [0, 1] required
    <config-format-xml> [0, 1] required
<kill-session> [0, 1] required
  <session-id> [0, 1] required
<lock> [0, 1] required
  <target> 1 required
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
<unlock> [0, 1] required
  <target> 1 required
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
<validate> [0, 1] required
  <source> 1 required
    <config> [0, 1] required
      <cli-config-data> [0, 1] required
        <cmd> 1+ required
      <cli-config-data-block> [0, 1] required
      <xml-config-data> [0, 1] required
        <Device-Configuration> [0, 1] required
          <> any subtree is allowed
    <candidate> [0, 1] required
    <running> [0, 1] required
```

```
    <startup> [0, 1] required
    <url> [0, 1] required
<notification-on> [0, 1] required
<notification-off> [0, 1] required
```

Additional References for NETCONF

Related Documents

Related Topic	Document Title
Cicso IOS commands	Cisco IOS Master Command List, All Releases
NETCONF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Cisco Networking Services Command Reference*
Security and IP access lists commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*

Standards and RFCs

Standard/RFC	Title
RFC 4251	*The Secure Shell (SSH) Protocol Architecture*
RFC 4252	*The Secure Shell (SSH) Authentication Protocol*
RFC 4741	*NETCONF Configuration Protocol*
RFC 4744	*Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NETCONF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information for NETCONF

Feature Name	Releases	Feature Information
NETCONF		The NETCONF protocol defines a simple mechanism through which a network device can be managed, configuration data can be retrieved, and new configuration data can be uploaded and manipulated. NETCONF uses Extensible Markup Language (XML)-based data encoding for the configuration data and protocol messages. The following commands were introduced or modified by this feature: **clear netconf**, **debug netconf**, **show netconf**.
NETCONF XML PI		The NETCONF protocol was enhanced, adding format attribute support for all Cisco IOS exec commands. The following commands were modified: **clear netconf**, **debug netconf**, and **show netconf**.

Glossary

BEEP —Blocks Extensible Exchange Protocol. A generic application protocol framework for connection-oriented, asynchronous interactions.

NETCONF —Network Configuration Protocol. A protocol that defines a simple mechanism through which a network device can be managed, configuration data can be retrieved, and new configuration data can be uploaded and manipulated.

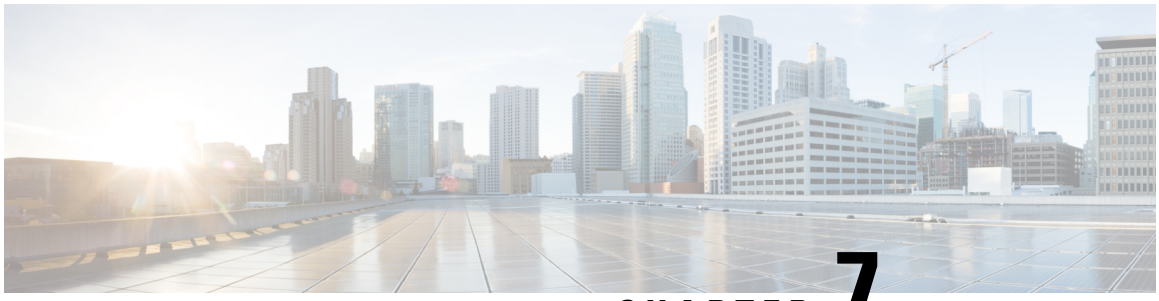
SASL —Simple Authentication and Security Layer. An Internet standard method for adding authentication support to connection-based protocols. SASL can be used between a security appliance and a Lightweight Directory Access Protocol (LDAP) server to secure user authentication.

SSHv2 —Secure Shell Version 2. SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSHv2 provides a means to securely access and securely execute commands on another computer over a network.

TLS —Transport Layer Security. An application-level protocol that provides for secure communication between a client and server by allowing mutual authentication, the use of hash for integrity, and encryption for privacy. TLS relies upon certificates, public keys, and private keys.

XML —Extensible Markup Language. A standard maintained by the World Wide Web Consortium (W3C) that defines a syntax that lets you create markup languages to specify information structures. Information structures define the type of information (for example, subscriber name or address), not how the information

appears (bold, italic, and so on). External processes can manipulate these information structures and publish them in a variety of formats. XML allows you to define your own customized markup language.



CHAPTER 7

NETCONF over SSHv2

You can use the Network Configuration Protocol (NETCONF) over Secure Shell Version 2 (SSHv2) feature to perform network configurations via the Cisco command-line interface (CLI) over an encrypted transport. The NETCONF Network Manager, which is the NETCONF client, must use Secure Shell Version 2 (SSHv2) as the network transport to the NETCONF server. Multiple NETCONF clients can connect to the NETCONF server.

Prerequisites for NETCONF over SSHv2

- NETCONF over SSHv2 requires that a vty line be available for each NETCONF session as specified in the **netconf max-session** command.

Restrictions for NETCONF over SSH

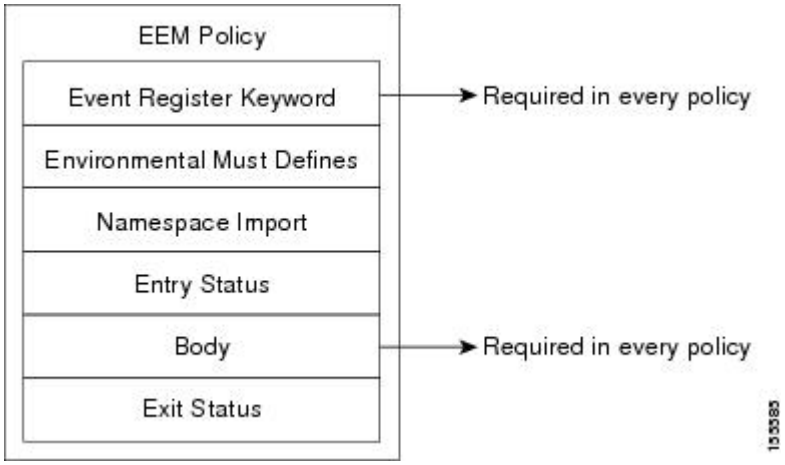
- Network Configuration Protocol (NETCONF) Secure Shell Version 2 (SSHv2) supports a maximum of 16 concurrent sessions.
- Only SSH version 2 is supported.

Information About NETCONF over SSHv2

NETCONF over SSHv2

To run the NETCONF over SSHv2 feature, the client (a Cisco device running Cisco software) establishes an SSH transport connection with the server (a NETCONF network manager). The following image shows a basic NETCONF over SSHv2 network configuration. The client and server exchange keys for security and password encryption. The user ID and password of the SSHv2 session running NETCONF are used for authorization and authentication purposes. The user privilege level is enforced and the client session may not have full access to the NETCONF operations if the privilege level is not high enough. If authentication, authorization, and accounting (AAA) is configured, the AAA service is used as if a user had established an SSH session directly to the device. Using the existing security configuration makes the transition to NETCONF almost seamless. Once the client has been successfully authenticated, the client invokes the SSH connection protocol and the SSH session is established. After the SSH session is established, the user or application invokes NETCONF as an SSH subsystem called “netconf.”

Figure 1: NETCONF over SSHv2



Secure Shell Version 2

SSHv2 runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSHv2 provides a means to securely access and securely execute commands on another computer over a network.

NETCONF does not support SSH version 1. The configuration for the SSH Version 2 server is similar to the configuration for SSH version 1. Use the **ip ssh version** command to specify which version of SSH that you want to configure. If you do not configure this command, SSH by default runs in compatibility mode; that is, both SSH version 1 and SSH version 2 connections are honored.

Note SSH version 1 is a protocol that has never been defined in a standard. If you do not want your device to fall back to the undefined protocol (version 1), you should use the **ip ssh version** command and specify version 2.

Use the **ip ssh rsa keypair-name** command to enable an SSH connection using Rivest, Shamir, and Adelman (RSA) keys that you have configured. If you configure the **ip ssh rsa keypair-name** command with a key-pair name, SSH is enabled if the key pair exists, or SSH will be enabled if the key pair is generated later. If you use this command to enable SSH, you do not need to configure a hostname and a domain name.

How to Configure NETCONF over SSHv2

Enabling SSH Version 2 Using a Hostname and Domain Name

Perform this task to configure your device for SSH version 2 using a hostname and domain name. You may also configure SSH version 2 by using the RSA key pair configuration (see Enabling SSH Version 2 Using RSA Key Pairs, on page 64).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*]
7. **ip ssh version 2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *hostname* **Example:** `Device(config)# hostname host1`	Configures a hostname for your device.
Step 4	**ip domain-name** *name* **Example:** `Device(config)# ip domain-name domain1.com`	Configures a domain name for your device.

	Command or Action	Purpose
Step 5	**crypto key generate rsa** **Example:** `Device(config)# crypto key generate rsa`	Enables the SSH server for local and remote authentication.
Step 6	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Device(config)# ip ssh timeout 120`	(Optional) Configures SSH control variables on your device.
Step 7	**ip ssh version 2** **Example:** `Device(config)# ip ssh version 2`	Specifies the version of SSH to be run on your device.

Enabling SSH Version 2 Using RSA Key Pairs

Perform this task to enable SSH version 2 without configuring a hostname or domain name. SSH version 2 will be enabled if the key pair that you configure already exists or if it is generated later. You may also configure SSH version 2 by using the hostname and domain name configuration. (See "Enabling SSH Version 2 Using a Hostname and Domain Name, on page 63.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh rsa keypair-name** *keypair-name*
4. **crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size*
5. **ip ssh** [**timeout** *seconds* \| **authentication-retries** *integer*]
6. **ip ssh version 2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh rsa keypair-name** *keypair-name*	Specifies which RSA keypair to use for SSH usage.

	Command or Action	Purpose
	Example: `Device(config)# ip ssh rsa keypair-name sshkeys`	**Note** A Cisco device can have many RSA key pairs.
Step 4	**crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size* **Example:** `Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768`	Enables the SSH server for local and remote authentication on the device. For SSH version 2, the modulus size must be at least 768 bits. **Note** To delete the RSA key pair, use the **crypto key zeroize rsa** command. After you have deleted the RSA command, you automatically disable the SSH server.
Step 5	**ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*] **Example:** `Device(config)# ip ssh timeout 120`	Configures SSH control variables on your device.
Step 6	**ip ssh version 2** **Example:** `Device(config)# ip ssh version 2`	Specifies the version of SSH to be run on a device.

Starting an Encrypted Session with a Remote Device

Perform this task to start an encrypted session with a remote networking device. (You do not have to enable your device. SSH can be run in disabled mode.)

From any UNIX or UNIX-like device, the following command is typically used to form an SSH session:

```
ssh -2 -s user@router.example.com netconf
```

SUMMARY STEPS

1. Do one of the following:
   - **ssh** [**-v** {**1** | **2**}] [**-c** {**3des**| **aes128-cbc** | **aes192-cbc**| **aes256-cbc**}] [**-m**{**hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Do one of the following: • **ssh** [**-v** {**1** | **2**}] [**-c** {**3des**| **aes128-cbc** | **aes192-cbc**| **aes256-cbc**}] [**-m**{**hmac-md5** | **hmac-md5-96** |	Starts an encrypted session with a remote networking device.

	Command or Action	Purpose

	hmac-sha1 | **hmac-sha1-96**}] [**l** *userid*] [**-o numberofpasswordprompts** *n*] [**-p** *port-num*] {*ip-addr* | *hostname*} [*command*]
	Example:
	`Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24`
	Example:
	`Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 user2@10.76.82.24`
		The first example adheres to the SSH version 2 conventions. A more natural and common way to start a session is by linking the username with the hostname. For example, the second configuration example provides an end result that is identical to that of the first example.

Troubleshooting Tips

The **ip ssh version** command can be used for troubleshooting your SSH configuration. By changing versions, you can determine which SSH version has a problem.

What to Do Next

For more information about the **ssh** command, see the Cisco IOS Security Command Reference.

Verifying the Status of the Secure Shell Connection

Perform this task to display the status of the SSH connection on your device.

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ssh**
3. **show ip ssh**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ssh** **Example:** `Device# show ssh`	Displays the status of SSH server connections.
Step 3	**show ip ssh** **Example:** `Device# show ip ssh`	Displays the version and configuration data for SSH.

Examples

The following output from the **show ssh** command displays status about SSH version 2 connections.

```
Device# show ssh
Connection Version Mode Encryption  Hmac                State
Username
1          2.0     IN   aes128-cbc  hmac-md5     Session started       lab
1          2.0     OUT  aes128-cbc  hmac-md5     Session started       lab
%No SSHv1 server connections running.
```

The following output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries.

```
Device# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

Enabling NETCONF over SSHv2

Perform this task to enable NETCONF over SSHv2.

Before you begin

SSHv2 must be enabled.

Note There must be at least as many vty lines configured as there are concurrent NETCONF sessions.

Note

- A minimum of four concurrent NETCONF sessions must be configured.
- A maximum of 16 concurrent NETCONF sessions can be configured.
- NETCONF does not support SSHv1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **netconf ssh** [**acl** *access-list-number*]
4. **netconf lock-time** *seconds*
5. **netconf max-sessions** *session*
6. **netconf max-message** size

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**netconf ssh** [**acl** *access-list-number*] **Example:** `Device(config)# netconf ssh acl 1`	Enables NETCONF over SSHv2. • Optionally, you can configure an access control list for this NETCONF session.
Step 4	**netconf lock-time** *seconds* **Example:** `Device(config)# netconf lock-time 60`	(Optional) Specifies the maximum time, in seconds, a NETCONF configuration lock is in place without an intermediate operation. • The valid range is 1 to 300. The default value is 10 seconds.
Step 5	**netconf max-sessions** *session* **Example:** `Device(config)# netconf max-sessions 5`	(Optional) Specifies the maximum number of concurrent NETCONF sessions allowed. • The valid range is 4 to 16. The default value is 4.
Step 6	**netconf max-message** size **Example:** `Device(config)# netconf max-message 37283`	(Optional) Specifies the maximum size, in kilobytes (KB), for the messages received in a NETCONF session. • The valid range is 1 to 2147483. The default value is infinite. • To set the maximum size to infinite, use the **no netconf max-message** command.

Configuration Examples for NETCONF over SSHv2

Example: Enabling SSHv2 Using a Hostname and Domain Name

```
configure terminal
hostname host1
ip domain-name example.com
crypto key generate rsa
ip ssh timeout 120
ip ssh version 2
```

Enabling Secure Shell Version 2 Using RSA Keys Example

The following example shows how to configure SSHv2 using RSA keys:

```
Device# configure terminal

Device(config)# ip ssh rsa keypair-name sshkeys

Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768
Device(config)# ip ssh timeout 120
Device(config)# ip ssh version 2
```

Starting an Encrypted Session with a Remote Device Example

The following example shows how to start an encrypted SSH session with a remote networking device, from any UNIX or UNIX-like device:

```
Device(config)# ssh -2 -s user@router.example.com netconf
```

Configuring NETCONF over SSHv2 Example

The following example shows how to configure NETCONF over SSHv2:

```
Device# configure terminal
Device(config)# netconf ssh acl 1
Device(config)# netconf lock-time 60
Device(config)# netconf max-sessions 5
Device(config)# netconf max-message 2345
Device# ssh-2 -s username@10.1.1.1 netconf
```

The following example shows how to get the configuration for loopback interface 113.

SUMMARY STEPS

1. First, send the “hello”:
2. Next, send the get-config request:

DETAILED STEPS

	Command or Action	Purpose
Step 1	First, send the “hello”: **Example:** <?xml version="1.0" encoding=\"UTF-8\"?> <hello><capabilities> <capability>u?rn:ietf:params:netconf:base:1.0</capability> <capability>urn:ietf:params:netconf:capability:writeable-running:1.0</capability> <capability>urn:ietf:params:netconf:ca?pability:roll?back-on-error:1.0</capability> <capability>urn:ietf:params:netconf:capability:startup:1.0</capability> <capability>urn:ietf:params:netconf:ca?pability:url:?1.0</capability> <capability>urn:cisco:params:netconf:capability:pi-data-model:1.0</capability> <capability>urn:cisco:params:netconf:capabili?ty:notificati?on:1.0</capability> </capabilities> </hello>]]>]]>	
Step 2	Next, send the get-config request: **Example:** <?xml version="1.0"?> <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"xmlns:cpi="http://www.cisco.com/cpi_10/schema" message-id="101"> <get-config> <source> <running/> </source> <filter> <config-format-text-cmd> <text-filter-spec> interface Loopback113 </text-filter-spec> </config-format-text-cmd> </filter> </get-config> </rpc>]]>]]>	

The following output is shown on the device:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101"xmlns=\"urn:ietf:params:netconf:base:1.0\">
```

```
   <data>
      <cli-config-data>
interface Loopback113
description test456
no ip address
load-interval 30
end
       </cli-config-data>
    </data>
</rpc-rep?ly>]]>]]>
```

Additional References for NETCONF over SSHv2

Related Documents

Related Topic	Document Title
Cicso IOS Commands	Cisco IOS Master Command List, All Releases
NETCONF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Cisco Networking Services Command Reference*
IP access lists commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
IP access lists	IP Access List Overview and Creating an IP Access List and Applying It to an Interface modules in the Cisco IOS Security Configuration Guide: Securing the Data Plane.
Secure Shell and Secure Shell Version 2	"Configuring Secure Shell" module in the Cisco IOS Security Configuration Guide: Securing User Services.

Standards and RFCs

RFC	Title
RFC 2246	*The TLS Protocol Version 1.0*
RFC 4251	*The Secure Shell (SSH) Protocol Architecture*
RFC 4252	*The Secure Shell (SSH) Authentication Protocol*
RFC 4741	NETCONF Configuration Protocol
RFC 4742	Using the NETCONF Configuration Protocol over Secure SHell (SSH)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

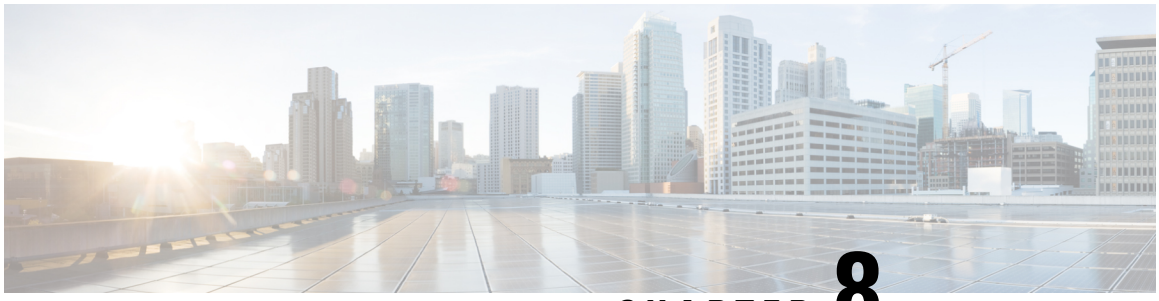
Feature Information for NETCONF over SSHv2

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for NETCONF over SSHv2

Feature Name	Releases	Feature Information
NETCONF over SSHv2	Cisco IOS XE Release 2.1 12.2(33)SB 12.2(33)SRA 12.2(33)SXI 12.4(9)T	The NETCONF over SSHv2 feature enables you to perform network configurations via the Cisco command-line interface (CLI) over an encrypted transport. The following commands were introduced or modified by this feature: **netconf lock-time**, **netconf max-message**, **netconf max-sessions** **netconf ssh**.



CHAPTER 8

NETCONF Access for Configurations over BEEP

You can use the Network Configuration Protocol (NETCONF) over Blocks Extensible Exchange Protocol (BEEP) feature to send notifications of any configuration change over NETCONF. A notification is an event indicating that a configuration change has happened. The change can be a new configuration, deleted configuration, or changed configuration. The notifications are sent at the end of a successful configuration operation as one message showing the set of changes, rather than individual messages for each line in the configuration that is changed.

BEEP can use the Simple Authentication and Security Layer (SASL) profile to provide simple and direct mapping to the existing security model. Alternatively, NETCONF over BEEP can use the transport layer security (TLS) to provide a strong encryption mechanism with either server authentication or server and client-side authentication.

Prerequisites for NETCONF Access for Configurations over BEEP

NETCONF over BEEP listeners require Simple Authentication and Security layer (SASL) to be configured.

Restrictions for NETCONF Access for Configurations over BEEP

You must be running a crypto image in order to configure BEEP using transport layer security (TLS).

Information About NETCONF Access for Configurations over BEEP

NETCONF over BEEP Overview

The NETCONF Access for Configurations over BEEP feature allows you to enable BEEP as the transport protocol to use during NETCONF sessions. Using NETCONF over BEEP, you can configure either the NETCONF server or the NETCONF client to initiate a connection, thus supporting large networks of intermittently connected devices, and those devices that must reverse the management connection where there are firewalls and Network Address Translators (NATs).

BEEP is a generic application protocol framework for connection-oriented, asynchronous interactions. It is intended to provide the features that traditionally have been duplicated in various protocol implementations. BEEP typically runs on top of Transmission Control Protocol (TCP) and allows the exchange of messages. Unlike HTTP and similar protocols, either end of the connection can send a message at any time. BEEP also includes facilities for encryption and authentication and is highly extensible.

The BEEP protocol contains a framing mechanism that permits simultaneous and independent exchanges of messages between peers. These messages are usually structured using XML. All exchanges occur in the context of a binding to a well-defined aspect of the application, such as transport security, user authentication, or data exchange. This binding forms a channel; each channel has an associated profile that defines the syntax and semantics of the messages exchanged.

The BEEP session is mapped onto the NETCONF service. When a session is established, each BEEP peer advertises the profiles it supports. During the creation of a channel, the client (the BEEP initiator) supplies one or more proposed profiles for that channel. If the server (the BEEP listener) creates the channel, it selects one of the profiles and sends it in a reply. The server may also indicate that none of the profiles are acceptable, and decline creation of the channel.

BEEP allows multiple data exchange channels to be simultaneously in use.

Although BEEP is a peer-to-peer protocol, each peer is labeled according to the role it is performing at a given time. When a BEEP session is established, the peer that awaits new connections is the BEEP listener. The other peer, which establishes a connection to the listener, is the BEEP initiator. The BEEP peer that starts an exchange is the client, and the other BEEP peer is the server. Typically, a BEEP peer that acts in the server role also performs in the listening role. However, because BEEP is a peer-to-peer protocol, the BEEP peer that acts in the server role is not required to also perform in the listening role.

NETCONF over BEEP and SASL

The SASL is an Internet standard method for adding authentication support to connection-based protocols. SASL can be used between a security appliance and an Lightweight Directory Access Protocol (LDAP) server to secure user authentication.

BEEP listeners require SASL to be configured.

NETCONF over BEEP and TLS

The TLS is an application-level protocol that provides for secure communication between a client and server by allowing mutual authentication, the use of hash for integrity, and encryption for privacy. TLS relies upon certificates, public keys, and private keys.

Certificates are similar to digital ID cards. They prove the identity of the server to clients. Each certificate includes the name of the authority that issued it, the name of the entity to which the certificate was issued, the entity's public key, and time stamps that indicate the certificate's expiration date.

Public and private keys are the ciphers used to encrypt and decrypt information. Although the public key is shared, the private key is never given out. Each public-private key pair works together. Data encrypted with the public key can be decrypted only with the private key.

NETCONF over BEEP and Access Lists

You can optionally configure access lists for use with NETCONF over SSHv2 sessions. An access list is a sequential collection of permit and deny conditions that apply to IP addresses. The Cisco software tests addresses against the conditions in an access list one by one. The first match determines whether the software accepts or rejects the address. Because the software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the software rejects the address.

The two main tasks involved in using access lists are as follows:

1. Creating an access list by specifying an access list number or name and access conditions.
2. Applying the access list to interfaces or terminal lines.

For more information about configuring access lists, see "IP Access List Overview" and "Creating an IP Access List and Applying It to an Interface" modules in *Security Configuration Guide: Securing the Data Plane*.

How to Configure NETCONF Access for Configurations over BEEP

Configuring an SASL Profile

To enable NETCONF over BEEP using SASL, you must first configure an SASL profile, which specifies which users are allowed access into the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sasl profile** *profile-name*
4. **mechanism di gest-md5**
5. **server** *user-name* **password** *password*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**sasl profile** *profile-name* **Example:** `Device(config)# sasl profile beep`	Configures an SASL profile and enters SASL profile configuration mode.
Step 4	**mechanism di gest-md5** **Example:** `Device(config-SASL-profile)# mechanism digest-md5`	Configures the SASL profile mechanism.
Step 5	**server** *user-name* **password** *password* **Example:** `Device(config-SASL-profile)# server user1 password password1`	Configures an SASL server.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Enabling NETCONF over BEEP

Before you begin

- There must be at least as many vty lines configured as there are concurrent NETCONF sessions.
- If you configure NETCONF over BEEP using SASL, you must first configure an SASL profile.

Note
- A minimum of four concurrent NETCONF sessions must be configured.
- A maximum of 16 concurrent NETCONF sessions can be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa general-keys**

4. **crypto pki trustpoint** *name*
5. **enrollment url** *url*
6. **subject-name** *name*
7. **revocation-check** *method1* [*method2* [*method3*]]
8. **exit**
9. **crypto pki authenticate** *name*
10. **crypto pki enroll** *name*
11. **netconf lock-time** *seconds*
12. **line vty** line-number [*ending-line-number*]
13. **netconf max-sessions** *session*
14. **netconf beep initiator** {*hostname* | *ip-address*} *port-number* **user** *sasl-user* **password** *sasl-password*[**encrypt** *trustpoint*] [**reconnect-time** *seconds*]
15. **netconf beep listener** [*port-number*] [**acl** *access-list-number*] [**sasl** *sasl-profile*] [**encrypt** *trustpoint*]
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto key generate rsa general-keys** **Example:** `Device(config)# crypto key generate rsa general-keys`	Generates Rivest, Shamir, and Adelman (RSA) key pairs and specifies that the general-purpose key pair should be generated. Perform this step only once.
Step 4	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint my_trustpoint`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 5	**enrollment url** *url* **Example:** `Device(ca-trustpoint)# enrollment url http://10.2.3.3:80`	Specifies the enrollment parameters of a certification authority (CA).
Step 6	**subject-name** *name* **Example:**	Specifies the subject name in the certificate request.

	Command or Action	Purpose
	`Device(ca-trustpoint)# subject-name CN=dns_name_of_host.com`	**Note** The subject name should be the Domain Name System (DNS) name of the device.
Step 7	**revocation-check** *method1* [*method2* [*method3*]] **Example:** `Device(ca-trustpoint)# revocation-check none`	Checks the revocation status of a certificate.
Step 8	**exit** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 9	**crypto pki authenticate** *name* **Example:** `Device(config)# crypto pki authenticate my_trustpoint`	Authenticates the certification authority (by getting the certificate of the CA).
Step 10	**crypto pki enroll** *name* **Example:** `Device(config)# crypto pki enroll my_trustpoint`	Obtains the certificate or certificates for your router from CA.
Step 11	**netconf lock-time** *seconds* **Example:** `Device(config)# netconf lock-time 60`	(Optional) Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation. The valid value range for the seconds argument is 1 to 300 seconds. The default value is 10 seconds.
Step 12	**line vty** line-number [*ending-line-number*] **Example:** `Device(config)# line vty 0 15`	Identifies a specific virtual terminal line for remote console access. You must configure the same number of vty lines as maximum NETCONF sessions.
Step 13	**netconf max-sessions** *session* **Example:** `Device(config)# netconf max-sessions 16`	(Optional) Specifies the maximum number of concurrent NETCONF sessions allowed.
Step 14	**netconf beep initiator** {*hostname* | *ip-address*} *port-number* **user** *sasl-user* **password** *sasl-password*[**encrypt** *trustpoint*] [**reconnect-time** *seconds*] **Example:** `Device(config)# netconf beep initiator host1 23`	(Optional) Specifies BEEP as the transport protocol for NETCONF sessions and configures a peer as the BEEP initiator. **Note** Perform this step to configure a NETCONF BEEP initiator session. You can also optionally configure a BEEP listener session.

	Command or Action	Purpose
	`user user1 password password1 encrypt 23 reconnect-time 60`	
Step 15	**netconf beep listener** [*port-number*] [**acl** *access-list-number*] [**sasl** *sasl-profile*] [**encrypt** *trustpoint*] **Example:** `Device(config)# netconf beep listener 26 acl 101 sasl profile1 encrypt 25`	(Optional) Specifies BEEP as the transport protocol for NETCONF and configures a peer as the BEEP listener. **Note** Perform this step to configure a NETCONF BEEP listener session. You can also optionally configure a BEEP initiator session.
Step 16	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for NETCONF Access for Configurations over BEEP

Example: Enabling NETCONF over BEEP

```
Device# configure terminal
Device(config)# crypto key generate rsa general-keys

Device(ca-trustpoint)# crypto pki trustpoint my_trustpoint

Device(ca-trustpoint)# enrollment url http://10.2.3.3:80
Device(ca-trustpoint)# subject-name CN=dns_name_of_host.com
Device(ca-trustpoint)# revocation-check none
Device(ca-trustpoint)# crypto pki authenticate my_trustpoint

Device(ca-trustpoint)# crypto pki enroll my_trustpoint

Device(ca-trustpoint)# line vty 0 15

Device(ca-trustpoint)# exit
Device(config)# netconf lock-time 60

Device(config)# netconf max-sessions 16

Device(config)# netconf beep initiator host1 23 user my_user password my_password encrypt
my_trustpoint reconnect-time 60

Device(config)# netconf beep listener 23 sasl user1 encrypt my_trustpoint
```

Additional References for NETCONF Access for Configurations over BEEP

Related Documents

Related Topic	Document Title
Cicso IOS Commands	Cisco IOS Master Commands List, All Releases
NETCONF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Cisco Networking Services Command Reference*

Standards and RFCs

Standard/RFC	Title
RFC 2222	*Simple Authentication and Security Layer (SASL)*
RFC 3080	*The Blocks Extensible Exchange Protocol Core*
RFC 4741	*NETCONF Configuration Protocol*
RFC 4744	*Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NETCONF Access for Configurations over BEEP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

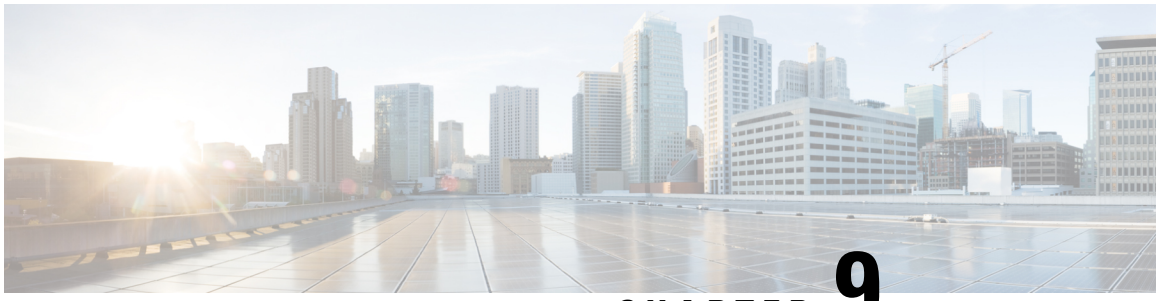
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 9: Feature Information for NETCONF Access for Configurations over BEEP

Feature Name	Releases	Feature Information
NETCONF Access for Configurations over BEEP	Cisco IOS XE Release 2.1 12.2(33)SB 12.2(33)SRB 12.2(33)SXI 12.4(9)T	The NETCONF over BEEP feature allows you to enable either the NETCONF server or the NETCONF client to initiate a connection, thus supporting large networks of intermittently connected devices and those devices that must reverse the management connection where there are firewalls and network address translators (NATs). The following commands were introduced or modified by this feature: **netconf beep initiator**, **netconf beep listener**.

PART II

Network Management



CHAPTER 9

Cisco IOS XE Scripting with Tcl

The Cisco IOS XE Scripting with Tcl feature provides the ability to run Tool Command Language (Tcl) version 8.3.4 commands from the Cisco IOS XE command-line interface (CLI).

Prerequisites for Cisco IOS XE Scripting with Tcl

- Familiarity with Tcl programming and Cisco IOS XE commands is assumed.
- Tcl commands can be executed from the Tcl configuration mode using the Cisco IOS XE CLI. Tcl configuration mode, like global configuration mode, is accessed from privileged EXEC mode. Access to privileged EXEC mode should be managed by restricting access using the **enable** command password.

Restrictions for Cisco IOS XE Scripting with Tcl

- If Cisco IOS XE configuration commands are used within the Tcl scripts, submode commands must be entered as quoted arguments on the same line as the configuration command.
- Error messages are provided, but you must check that the Tcl script will run successfully because errors may cause the Tcl shell to run in an infinite loop.

Caution The use of Tcl server sockets to listen to telnet and FTP ports (23 and 21 respectively) will preempt the normal handling of these ports in Cisco IOS XE software.

- The table below lists Tcl commands and library calls that do not behave within Cisco IOS XE software as documented in standard Tcl documents.

Table 10: Tcl Command Options That Behave Differently in Cisco IOS XE Software

Command	Keyword	Argument	Supported	Comments
after	**ms**	*script*	Partially	When the CLI **tclsh** command is used, there is no event loop implemented unless Embedded Syslog Manager (ESM) is active on the same router. Commands entered using the **after** Tcl command will not run unless forced using the **update** command. Sleep mode (the **after** command) works only with the **ms** keyword.
file	**-time**	*atime*	No	The optional **-time** keyword to set the file access time is not supported in Cisco IOS XE software.
file	**-time**	*mtime*	No	The optional **-time** keyword to set the file modification time is not supported in Cisco IOS XE software.
fileevent			Partially	When the CLI **tclsh** command is used, there is no event loop implemented unless Embedded Syslog Manager (ESM) is active on the same router. Commands entered using the **fileevent** Tcl command will not run unless forced using the **update** command.
history	**!** *n*		Partially	The **!** *n* shortcut does not work in Cisco IOS XE software. Use the **history** Tcl command with the **redo** *n* keyword.
load			No	When the CLI **load** command is used, an error message stating "dynamic loading not available on this system" is displayed.

Information About Cisco IOS XE Scripting with Tcl

Tcl Shell for Cisco IOS XE Software

The Cisco IOS XE Tcl shell was designed to allow customers to run Tcl commands directly from the Cisco IOS XE CLI prompt. Cisco IOS XE software does contain some subsystems such as Embedded Syslog Manager (ESM) and Interactive Voice Response (IVR) that use Tcl interpreters as part of their implementation. These subsystems have their own proprietary commands and keyword options that are not available in the Tcl shell.

Several methods have been developed for creating and running Tcl scripts within Cisco IOS XE software. A Tcl shell can be enabled, and Tcl commands can be entered line by line. After Tcl commands are entered, they are sent to a Tcl interpreter. If the commands are recognized as valid Tcl commands, the commands are executed and the results are sent to the tty. If a command is not a recognized Tcl command, it is sent to the Cisco IOS XE CLI parser. If the command is not a Tcl or Cisco IOS XE command, two error messages are displayed. A predefined Tcl script can be created outside of Cisco IOS XE software, transferred to flash or disk memory, and run within Cisco IOS XE software. It is also possible to create a Tcl script and precompile the code before running it under Cisco IOS XE software.

Multiple users on the same router can be in Tcl configuration mode at the same time without interference because each Tcl shell session launches a separate interpreter and Tcl server process. The tty interface number

served by each Tcl process is represented in the server process name and can be displayed using the **show process** CLI command.

The Tcl shell can be used to run Cisco IOS XE CLI EXEC commands within a Tcl script. Using the Tcl shell to run CLI commands allows customers to build menus to guide novice users through tasks, to automate repetitive tasks, and to create custom output for **show** commands.

Tcl Precompiler

The Cisco IOS XE Tcl implementation offers support for loading scripts that have been precompiled by the TclPro precompiler. Precompiled scripts allow a measure of security and consistency because they are obfuscated.

SNMP MIB Object Access

Designed to make access to Simple Network Management Protocol (SNMP) MIB objects easier, a set of UNIX-like SNMP commands has been created. The Tcl shell is enabled either manually or by using a Tcl script, and the new commands can be entered to allow you to perform specified get and set actions on MIB objects. To increase usability, the new commands have names similar to those used for UNIX SNMP access. To access the SNMP commands go to the Using the Tcl Shell to Access SNMP MIB Objects.

Custom Extensions in the Tcl Shell

The Cisco IOS XE implementation of the Tcl shell contains some custom command extensions. These extensions operate only under Tcl configuration mode. The table below displays these command extensions.

Table 11: Cisco IOS XE Custom Tcl Command Extensions

Command	Description
ios_config	Runs a Cisco IOS XE CLI configuration command.
log_user	Toggles Tcl command output under Tcl configuration mode.
typeahead	Writes text to the router standard input (stdin) buffer file.
tclquit	Leave Tcl shell--synonym for **exit**.

SNMP MIB Custom Extensions in the Tcl Shell

The Cisco IOS XE implementation of the Tcl shell contains some custom command extensions for SNMP MIB object access. These extensions operate only under Tcl configuration mode. The table below displays these command extensions.

Table 12: Cisco IOS XE Custom Tcl Command Extensions for SNMP MIB Access

Command	Description
snmp_getbulk	Retrieves a large section of a MIB table. This command is similar to the SNMP **getbulk** command. The syntax is in the following format: **snmp_getbulk** *community-string non-repeaters max-repetitions oid* [*oid2 oid3...*] • Use the *community-string* argument to specify the SNMP community from which the objects will be retrieved. • Use the *non-repeaters* argument to specify the number of objects that can be retrieved with a get-next operation. • Use the *max-repetitions* argument to specify the maximum number of get-next operations to attempt while trying to retrieve the remaining objects. • Use the *oid* argument to specify the object ID(s) to retrieve.
snmp_getid	Retrieves the following variables from the SNMP entity on the router: • sysDescr.0 • sysObjectID.0 • sysUpTime.0 • sysContact.0 • sysName.0 • sysLocation.0 This command is similar to the SNMP **getid** command. The syntax is in the following format: **snmp_getid** *community-string*
snmp_getnext	Retrieves a set of individual variables from the SNMP entity on the router. This command is similar to the SNMP **getnext**command. The syntax is in the following format: **snmp_getnext** *community-string oid* [*oid2 oid3...*]
snmp_getone	Retrieves a set of individual variables from the SNMP entity on the router. This command is similar to the SNMP **getone** command. The syntax is in the following format: **snmp_getone** *community-string oid* [*oid2 oid3...*]

Command	Description
snmp_setany	Retrieves the current values of the specified variables and then performs a set request on the variables. This command is similar to the SNMP **setany** command. The syntax is in the following format: **snmp_setany** *community-string oid type val* [*oid2 type2 val2...*] • Use the *type* argument to specify the type of object to retrieve. The *type* can be one of the following: • **-i**--Integer. A 32-bit number used to specify a numbered type within the context of a managed object. For example, to set the operational status of a router interface, 1 represents up and 2 represents down. • **-u**--Unsigned32. A 32-bit number used to represent decimal values in the range from 0 to 2 32 - 1 inclusive. • **-c**--Counter32. A 32-bit number with a minimum value of 0 and a maximum value of 2 32 - 1. When the maximum value is reached, the counter resets to 0 and starts again. • **-g**--Gauge. A 32-bit number with a minimum value of 0 and a maximum value of 2 32 - 1. The number can increase or decrease at will. For example, the interface speed on a router is measured using a gauge object type. • **-o**--Octet string. An octet string--in hex notation--used to represent physical addresses. • **-d**--Display string. An octet string--in text notation--used to represent text strings. • **-ipv4**--IP version 4 address. • **-oid**--Object ID. • Use the *val* argument to specify the value of object ID(s) to retrieve.

How to Configure Cisco IOS XE Scripting with Tcl

Enabling the Tcl Shell and Using the CLI to Enter Commands

Perform this task to enable the interactive Tcl shell and to enter Tcl commands through the Cisco IOS CLI. The optional steps in this procedure include specifying a default location for encoding files and specifying an initialization script.

Step 1 **enable**

Example:

```
Router> enable
```

Enables the privileged EXEC mode. Enter your password, if prompted.

If you're encoding your files, or if you're using an initialization script, or both, perform steps 2 through 6 in this procedure. Else, go to step 7.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters the global configuration mode.

Step 3 **scripting tcl encdir** *location-url*

Example:

```
Router(config)# scripting tcl encdir tftp://10.18.117.23/enctcl/
```

Specifies the default location of external encoding files used by the Tcl **encoding** command.

Step 4 **scripting tcl init** *init-url*

Example:

```
Router(config)# scripting tcl init ftp://user:password@172.17.40.3/tclscript/initfiles3.tcl
```

Specifies an initialization script to run when you enable the Tcl shell.

Step 5 **scripting tcl low-memory** *bytes*

Example:

```
Router(config)# scripting tcl low-memory 33117513
```

Specifies a low memory mark for free memory for the Tcl-based applications. You can set the memory threshold anywhere 0–4294967295 bytes.

Note If the minimum free RAM drops below this threshold, TCL terminates the current script. This action prevents the Tcl interpreter from allocating too much RAM to avoid the router from crashing.

Step 6 **exit**

Example:

```
Router(config)# exit
```

Exits the global configuration mode and returns to the privileged EXEC mode.

Step 7 **tclsh**

Example:

```
Router# tclsh
```

Enables the interactive Tcl shell and enters the Tcl configuration mode.

Step 8 Enter the required Tcl command language syntax.

Example:

```
Router(tcl)# proc get_bri {}
```

The commands you enter in the Tcl configuration mode are sent first to the interactive Tcl interpreter. If the command isn't a valid Tcl command, it's then sent to the CLI parser.

Step 9 **ios_config "** *cmd* **" "** *cmd-option* **"**

Example:

```
Router(tcl)# ios_config "interface Ethernet 2/0" "no keepalive"
```

Modifies the router configuration using a Tcl script by specifying the Tcl command **ios_config** with CLI commands and options.

In this example, the first argument in quotes configures an Ethernet interface and enters the interface configuration mode. The second argument in quotes sets the keepalive option, which keeps the connection open for multiple servicing requests. If you entered these two CLI statements on separate Tcl command lines, the configuration does not work.

Step 10 **socket -myaddr** *addr* **-myport** *port* **-myvrf** *vrf-table-name host port*

Example:

```
Router(tcl)# socket -myaddr 10.4.9.34 -myport 12345 -myvrf testvrf 12346
```

Specifies the client socket and allows a TCL interpreter to connect via TCP over IPv4/IPv6 and opens a TCP network connection. You can specify a port and host to connect to; there must be a server to accept connections on this port.

- **-myaddr** *addr* – the domain name or the IP address of the client-side network interface. Use this option if the client machine has multiple network interfaces.
- **-myport** *port* -- the port number required for the client connection.
- **-myvrf** [*vrf_table_name*]-- the vrf table name. This option returns the local VRF table name for the specified socket. If you have not configured a VRF table for the given socket the system displays a TCL_ERROR. A "No VRF table configured" message is appended to the interpreter result.

Step 11 **socket** - *server* **-myaddr** *addr* **-myvrf** *vrf-table-name port*

Example:

```
Router(tcl)# socket -server test -myvrf testvrf 12348
```

Specifies the server socket and allows a TCL interpreter to connect via TCP over IPv4/IPv6 and opens a TCP network connection. If the port is zero, Cisco IOS allocates a free port to the server socket by using the **fconfigure** command to read the *-sock0* argument.

- **-myaddr** *addr* – the domain name or the IP address of the server-side network interface. Use this option if the client machine has multiple network interfaces.
- **-myvrf** *vrf* -- the vrf table name. This option returns the local VRF table name for the specified socket. If you do not configure the vrf table the command returns a TCL_ERROR. The system displays the "Cannot obtain VRF Table ID for VRF_table_name" message.

Step 12 **fconfigure** *channelname* - *remote* [*host port*] - *broadcast boolean* - **vrf**[*vrf_table_name*]

Example:

```
Router(tcl)# fconfigure sock1 -vrf vrf1 -remote [list 10.4.9.37 56009] -broadcast 1
```

Specifies the options in a channel.

- In case of UDP sockets that are created using the **udp_open** command, you can map the UDP socket to a VRF using the **fconfigure** command.

- This command also enables you to display the properties of the channel.
- *-broadcast* -- enables or disables broadcasting.

Step 13 **udp_open** *-ipv6 port*

Example:

```
Router(tcl)# udp_open -ipv6 56005
```

Opens a UDP socket.

If you specify a port, the UDP socket is opened on that port. Optionally, the system chooses a port and you can use the **fconfigure** command to obtain the port number. If you specify the *-ipv6* argument, the socket is opened specifying the AF_INET6 protocol family.

Step 14 **udp_peek** *sock* **-buffersize** *buffer-size*

Example:

```
Router(tcl)# udp_peek sock0 -buffersize 100
```

Enables peeking into a UDP socket.

- **-buffersize** *buffer-size* --specifies the buffersize.

Step 15 **exec "** *exec-cmd* **"**

Example:

```
Router(tcl)# exec "show interfaces"
```

(Optional) Executes the Cisco IOS CLI EXEC mode commands from a Tcl script by specifying the Tcl command exec with the CLI commands.

- In this example, the system displays the interface information for the router.

Step 16 **exit**

Example:

```
Router(tcl)# exit
```

Exits Tcl configuration mode and returns to privileged EXEC mode.

Examples

The following sample (partial) output shows information about Ethernet interface 0 on the router. The **show interfaces** command has been executed from Tcl configuration mode.

```
Router# tclsh
Router(tcl)# exec "show interfaces"
Ethernet 0 is up, line protocol is up
  Hardware is MCI Ethernet, address is 0000.0c00.750c (bia 0000.0c00.750c)
  Internet address is 10.108.28.8, subnet mask is 255.255.255.0
```

```
  MTU 1500 bytes, BW 10000 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 4:00:00
  Last input 0:00:00, output 0:00:00, output hang never
  Last clearing of "show interface" counters 0:00:00
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 2000 bits/sec, 4 packets/sec
     1127576 packets input, 447251251 bytes, 0 no buffer
     Received 354125 broadcasts, 0 runts, 0 giants, 57186* throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5332142 packets output, 496316039 bytes, 0 underruns
     0 output errors, 432 collisions, 0 interface resets, 0 restarts
.
.
.
```

Running Predefined Tcl Scripts

Perform this optional task to run a predefined Tcl script in Cisco IOS XE software.

Before you begin

Before performing this task, you must create a Tcl script that can run on Cisco IOS XE software. The Tcl script may be transferred to internal flash memory using any file system that the Cisco IOS XE file system (IFS) supports, including TFTP, FTP, and rcp. The Tcl script may also be sourced from a remote location.

SUMMARY STEPS

1. **enable**
2. **tclsh**
3. Enter the Tcl source command with the filename and path.
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**tclsh** **Example:** `Router# tclsh`	Enables the interactive Tcl shell and enters Tcl configuration mode.
Step 3	Enter the Tcl source command with the filename and path. **Example:** `Router(tcl)# source slot0:test.tcl`	Commands entered in Tcl configuration mode are sent first to the interactive Tcl interpreter. If the command is not a valid Tcl command, it is then sent to the CLI parser.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(tcl)# exit`	Exits Tcl configuration mode and returns to privileged EXEC mode.

Configuration Examples for Cisco IOS XE Scripting with Tcl

Tcl Script Using the show interfaces Command Example

Using the Tcl regular expression engine, scripts can filter specific information from **show** commands and present it in a custom format. The following is an example of filtering the **show interfaces** command output and creating a comma-separated list of BRI interfaces on the router:

```
tclsh
proc get_bri {} {
    set check ""
    set int_out [exec "show interfaces"]
    foreach int [regexp -all -line -inline "(^BRI\[0-9]/\[0-9])" $int_out] {
        if {![string equal $check $int]} {
            if {[info exists bri_out]} {
                append bri_out "," $int
             } else {
                 set bri_out $int
             }
             set check $int
        }
    }
    return $bri_out
}
```

Tcl Script for SMTP Support Example

The following Tcl script is useful for sending e-mail messages from a router.

```
##
## Place required comments here!!!
##
package provide sendmail 2.0
# Sendmail procedure for Support
namespace eval ::sendmail {
    namespace export initialize configure sendmessage sendfile
    array set ::sendmail::sendmail {
        smtphost    mailhub
        from        ""
        friendly    ""
    }
    proc configure {} {}
    proc initialize {smtphost from friendly} {
        variable sendmail
        if {[string length $smtphost]} then {
            set sendmail(smtphost) $smtphost
        }
```

```
        if {[string length $from]} then {
            set sendmail(from) $from
        }
        if {[string length $friendly]} then {
            set sendmail(friendly) $friendly
        }
    }
    proc sendmessage {toList subject body {tcl_trace 0}} {
        variable sendmail
        set smtphost $sendmail(smtphost)
        set from $sendmail(from)
        set friendly $sendmail(friendly)
        if {$trace} then {
            puts stdout "Connecting to $smtphost:25"
        }
        set sockid [socket $smtphost 25]
## DEBUG
set status [catch {
        puts $sockid "HELO $smtphost"
        flush $sockid
        set result [gets $sockid]
        if {$trace} then {
            puts stdout "HELO $smtphost\n\t$result"
        }
        puts $sockid "MAIL From:<$from>"
        flush $sockid
        set result [gets $sockid]
        if {$trace} then {
            puts stdout "MAIL From:<$from>\n\t$result"
        }
        foreach to $toList {
            puts $sockid "RCPT To:<$to>"
            flush $sockid
        }
        set result [gets $sockid]
        if {$trace} then {
            puts stdout "RCPT To:<$to>\n\t$result"
        }
        puts  $sockid "DATA "
        flush $sockid
        set result [gets  $sockid]
        if {$trace} then {
            puts stdout "DATA \n\t$result"
        }
        puts  $sockid "From: $friendly <$from>"
        foreach to $toList {
            puts $sockid "To:<$to>"
        }
        puts  $sockid "Subject: $subject"
        puts  $sockid "\n"
        foreach line [split $body "\n"] {
            puts  $sockid " $line"
        }
        puts  $sockid "."
        puts  $sockid "QUIT"
        flush $sockid
        set result [gets  $sockid]
        if {$trace} then {
            puts stdout "QUIT\n\t$result"
        }
} result]
        catch {close $sockid }
        if {$status} then {
       return -code error $result
```

```
        }
        return
    }
    proc sendfile {toList filename subject {tcl_trace 0}} {
        set fd [open $filename r]
        sendmessage $toList $subject [read $fd] $trace
        return
    }
}
```

Tcl Script for SNMP MIB Access Examples

Using the Tcl shell, Tcl commands can perform actions on MIBs. The following example shows how to set up the community access strings to permit access to SNMP. Public access is read-only, but private access is read-write. The following example shows how to retrieve a large section of a table at once using the **snmp_getbulk** Tcl command extension.

Two arguments, *non-repeaters*and *max-repetitions*, must be set when an **snmp_getbulk** command is issued. The *non-repeaters* argument specifies that the first N objects are to be retrieved with a simple **snmp_getnext** operation. The *max-repetitions* argument specifies that up to M **snmp_getnext** operations are to be attempted to retrieve the remaining objects.

In this example, three bindings--sysUpTime (1.3.6.1.2.1.1.2.0), ifDescr (1.3.6.1.2.1.2.2.1.2), and ifType (1.3.6.1.2.1.2.2.1.3)--are used. The total number of variable bindings requested is given by the formula N + (M * R), where N is the number of non-repeaters (in this example 1), M is the max-repetitions (in this example 5), and R is the number of request objects (in this case 2, ifDescr and ifType). Using the formula, 1 + (5 * 2) equals 11; and this is the total number of variable bindings that can be retrieved by this **snmp_getbulk** request command.

Sample results for the individual variables include a retrieved value of sysUpTime.0 being 1336090, where the unit is in milliseconds. The retrieved value of ifDescr.1 (the first interface description) is FastEthernet0/0, and the retrieved value of ifType.1 (the first interface type) is 6, which corresponds to the ethernetCsmacd type.

```
snmp-server community public RO
snmp-server community private RW
tclsh
  snmp_getbulk public 1 5 1.3.6.1.2.1.1.2.0  1.3.6.1.2.1.2.2.1.2  1.3.6.1.2.1.2.2.1.3
  {<obj oid='sysUpTime.0' val='1336090'/>}
  {<obj oid='ifDescr.1' val='FastEthernet0/0'/>}
  {<obj oid='ifType.1' val='6'/>}
  {<obj oid='ifDescr.2' val='FastEthernet1/0'/>}
  {<obj oid='ifType.2' val='6'/>}
  {<obj oid='ifDescr.3' val='Ethernet2/0'/>}
  {<obj oid='ifType.3' val='6'/>}
  {<obj oid='ifDescr.4' val='Ethernet2/1'/>}
  {<obj oid='ifType.4' val='6'/>}
  {<obj oid='ifDescr.5' val='Ethernet2/2'/>}
  {<obj oid='ifType.5' val='6'/>}
```

The following example shows how to retrieve the sysDescr.0, sysObjectID.0, sysUpTime.0, sysContact.0, sysName.0, and sysLocation.0 variables--in this example shown as system.1.0, system.2.0, system.3.0, system.4.0, system.5.0, and system.6.0--from the SNMP entity on the router using the **snmp_getid** Tcl command extension.

```
tclsh
 snmp_getid public
```

```
 {<obj oid='system.1.0' val='Cisco Internetwork Operating System Software
 Cisco IOS XE(tm) 7200 Software (C7200-IK9S-M), Experimental Version 12.3(20030507:225511)
 [geotpi2itd1 124]
 Copyright (c) 1986-2003 by Cisco Systems, Inc.
 Compiled Wed 21-May-03 16:16 by engineer'/>}
 {<obj oid='system.2.0' val='products.223'/>}
 {<obj oid='sysUpTime.0' val='6664317'/>}
 {<obj oid='system.4.0' val='1-800-553-2447 - phone the TAC'/>}
 {<obj oid='system.5.0' val='c7200.myCompany.com'/>}
 {<obj oid='system.6.0' val='Bldg 24, San Jose, CA'/>}
```

The following example shows how to retrieve a set of individual variables from the SNMP entity on the router using the **snmp_getnext** Tcl command extension:

```
snmp_getnext public 1.3.6.1.2.1.1.1.0 1.3.6.1.2.1.1.2.0
 {<obj oid='system.2.0' val='products.223'/>}
 {<obj oid='sysUpTime.0' val='6683320'/>}
```

The following example shows how to retrieve a set of individual variables from the SNMP entity on the router using the **snmp_getone** Tcl command extension:

```
snmp_getone public 1.3.6.1.2.1.1.1.0 1.3.6.1.2.1.1.2.0
 {<obj oid='system.1.0' val='Cisco Internetwork Operating System Software
 Cisco IOS XE(tm) 7200 Software (C7200-IK9S-M), Experimental Version 12.3(20030507:225511)
 [geotpi2itd1 124]
 Copyright (c) 1986-2003 by Cisco Systems, Inc.
 Compiled Wed 21-May-03 16:16 by engineer'/>}
 {<obj oid='system.2.0' val='products.223'/>}
```

The following example shows how to change something in the configuration of the router using the **snmp_setany** Tcl command extension. In this example, the hostname of the router is changed to TCLSNMP-HOST.

```
tclsh
 snmp_setany private 1.3.6.1.2.1.1.5.0 -d TCLSNMP-HOST
 {<obj oid='system.5.0' val='TCLSNMP-HOST'/>}
```

Additional References

The following sections provide references related to the Signed Tcl Scripts feature.

Related Documents

Related Topic	Document Title
Cisco PKI Overview: Understanding and Planning a PKI Implementing and Managing a PKI	*Security Configuration Guide, Release 12.4*
PKI commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	*Cisco IOS Security Command Reference, Release 12.4*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

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

Feature Information for Cisco IOS XE Scripting with Tcl

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13: Feature Information for Cisco IOS XE Scripting with Tcl

Feature Name	Releases	Feature Information
Cisco IOS XE Scripting with Tcl	Cisco IOS XE	The Cisco IOS XE Scripting with Tcl feature provides the ability to run Tool Command Language (Tcl) version 8.3.4 commands from the Cisco IOS XE command-line interface (CLI). In Cisco IOS XE Release 2.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **scripting tcl encdir**, **scripting tcl init**, **scripting tcl low-memory**, **tclquit**, **tclsh**.
Tcl SNMP MIB Access	Cisco IOS XE	The Tcl SNMP MIB Access feature introduces a set of UNIX-like SNMP commands to make access to Simple Network Management Protocol (SNMP) MIB objects easier.

Glossary

CA--certification authority. Service responsible for managing certificate requests and issuing certificates to participating IPsec network devices. This service provides centralized key management for the participating devices and is explicitly trusted by the receiver to validate identities and to create digital certificates.

certificates--Electronic documents that bind a user's or device's name to its public key. Certificates are commonly used to validate a digital signature.

CRL--certificate revocation list. Electronic document that contains a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when the certificate was issued and when it expires. A new CRL is issued when the current CRL expires.

IPsec--IP security

peer certificate--Certificate presented by a peer, which contains the peer's public key and is signed by the trustpoint CA.

PKI--public key infrastructure. System that manages encryption keys and identity information for components of a network that participate in secured communications.

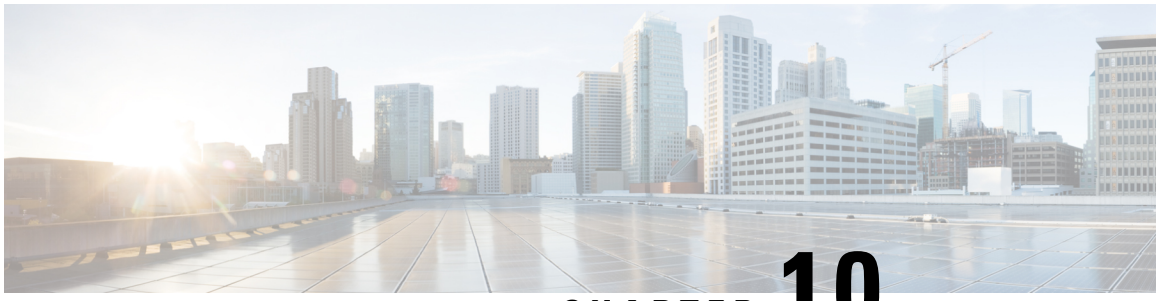
RA--registration authority. Server that acts as a proxy for the CA so that CA functions can continue when the CA is offline. Although the RA is often part of the CA server, the RA could also be an additional application, requiring an additional device to run it.

RSA keys--Public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. An RSA key pair (a public and a private key) is required before you can obtain a certificate for your device.

SHA1--Secure Hash Algorithm 1

SSH--secure shell

SSL--secure socket layer



CHAPTER 10

Embedded Packet Capture Overview

Embedded Packet Capture (EPC) provides an embedded systems management facility that helps in tracing and troubleshooting packets. This feature allows network administrators to capture data packets flowing through, to, and from a Cisco device. The network administrator may define the capture buffer size and type (circular, or linear), the maximum number of bytes of each packet to capture, and the direction of the traffic flow - ingress or egress, or both. The packet capture rate can be throttled using further administrative controls. For example, you can use the available options for filtering the packets to be captured using an Access Control List; and, optionally, further defined by specifying a maximum packet capture rate or by specifying a sampling interval.

Feature Information for Embedded Packet Capture

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 14: Feature Information for Embedded Packet Capture

Feature Name	Releases	Feature Information
Embedded Packet Capture support on LTE interface and FlexVPN interface	Cisco IOS XE Release Bengaluru 17.4.1a	Embedded Packet Capture (EPC) is an onboard packet capture facility that allows network administrators to capture packets flowing to, through, and from a device. With this feature, EPC is supported on LTE and FlexVPN interfaces. The following commands were introduced or modified: **show ip interface brief**, **show monitor capture epc**
Embedded Packet Capture	Cisco IOS XE Release 3.7S	Embedded Packet Capture (EPC) is an onboard packet capture facility that allows network administrators to capture packets flowing to, through, and from a device and to analyze them locally or save and export them for offline analysis using a tool such as Wireshark. This feature simplifies operations by allowing the devices to become active participants in the management and operation of the network. The following commands were introduced or modified: **debug epc**, **monitor capture (access list/class map)**, **monitor capture (interface/control plane)**, **monitor capture export**, **monitor capture limit**, **monitor capture start**, **monitor capture stop**, and **show monitor capture**.

Prerequisites for Embedded Packet Capture

The Embedded Packet Capture (EPC) software subsystem consumes CPU and memory resources during its operation. You must have adequate system resources for different types of operations. Some guidelines for using the system resources are provided in the table below.

Table 15: System Requirements for the EPC Subsystem

System Resources	Requirements
Hardware	CPU utilization requirements are platform dependent.
Memory	The packet buffer is stored in DRAM. The size of the packet buffer is user specified.
Diskspace	Packets can be exported to external devices. No intermediate storage on flash disk is requ

Restrictions for Embedded Packet Capture

- Embedded Packet Capture (EPC) captures multicast packets only on ingress and does not capture the replicated packets on egress.
- From Cisco IOS XE Release 3.7S, Embedded Packet Capture is only supported on Advance Enterprise Krypto (K9) images.
- From Cisco IOS XE Release 3.9S, Embedded Packet Capture is available on the following images:
  - IP Base Images
  - Special Services Images
  - Advance Security Images
  - Advance IP Services Images
  - Advance Enterprise Images

Information About Embedded Packet Capture

Embedded Packet Capture Overview

Embedded Packet Capture (EPC) provides an embedded systems management facility that helps in tracing and troubleshooting packets. This feature allows network administrators to capture data packets flowing through, to, and from a Cisco device. The network administrator may define the capture buffer size and type (circular, or linear), the maximum number of bytes of each packet to capture, and the direction of the traffic flow - ingress or egress, or both. The packet capture rate can be throttled using further administrative controls. For example, you can use the available options for filtering the packets to be captured using an Access Control List; and, optionally, further defined by specifying a maximum packet capture rate or by specifying a sampling interval.

Benefits of Embedded Packet Capture

- Ability to capture IPv4 and IPv6 packets in the device.
- Extensible infrastructure for enabling packet capture points. A capture point is a traffic transit point where a packet is captured and associated with a buffer.
- Facility to export the packet capture in packet capture file (PCAP) format suitable for analysis using any external tool.

- Methods to decode data packets captured with varying degrees of detail.

Packet Data Capture

Packet data capture is the capture of data packets that are then stored in a buffer. You can define packet data captures by providing unique names and parameters.

You can perform the following actions on the capture:

- Activate captures at any interface.
- Apply access control lists (ACLs) or class maps to capture points.

Note Network Based Application Recognition (NBAR) and MAC-style class map is not supported.

- Destroy captures.
- Specify buffer storage parameters such as size and type. The size ranges from 1 MB to 100 MB. The default buffer is linear; the other option for the buffer is circular.
- Specify any of the following limit options:
  - **duration** - limit total duration of capture in seconds.
  - **every** - limit capture to one in every nth packet.
  - **packet-len** - limit the packet length to capture.
  - **packets** - limit number of packets to capture.
  - **pps** - limit number of packets per second to capture.
- Specify match criteria that includes information about the protocol, IP address or port address.

How to Implement Embedded Packet Capture

Managing Packet Data Capture

SUMMARY STEPS

1. **enable**
2. **monitor capture** *capture-name* **access-list** *access-list-name*
3. **monitor capture** *capture-name* **limit duration** *seconds*
4. **monitor capture** *capture-name* **interface** *interface-name* **both**
5. **monitor capture** *capture-name* **buffer circular size** *bytes*
6. **monitor capture** *capture-name* **start**
7. **monitor capture** *capture-name* **export** *file-location/file-name*

8. **monitor capture** *capture-name* **stop**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**monitor capture** *capture-name* **access-list** *access-list-name* **Example:** `Device# monitor capture mycap access-list v4acl`	Configures a monitor capture specifying an access list as the core filter for the packet capture.
Step 3	**monitor capture** *capture-name* **limit duration** *seconds* **Example:** `Device# monitor capture mycap limit duration 1000`	Configures monitor capture limits.
Step 4	**monitor capture** *capture-name* **interface** *interface-name* **both** **Example:** `Device# monitor capture mycap interface GigabitEthernet 0/0/1 both`	Configures monitor capture specifying an attachment point and the packet flow direction. **Note** • To change the traffic direction from **both** to **in** (ingress direction), enter the **no monitor capture** *capture-name* **interface** *interface-name* **out** command. • To change the traffic direction from **both** to **out** (egress direction), enter the **no monitor capture** *capture-name* **interface** *interface-name* **in** command.
Step 5	**monitor capture** *capture-name* **buffer circular size** *bytes* **Example:** `Device# monitor capture mycap buffer circular size 10`	Configures a buffer to capture packet data.
Step 6	**monitor capture** *capture-name* **start** **Example:** `Device# monitor capture mycap start`	Starts the capture of packet data at a traffic trace point into a buffer.
Step 7	**monitor capture** *capture-name* **export** *file-location/file-name* **Example:** `Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap`	Exports captured data for analysis.

	Command or Action	Purpose
Step 8	**monitor capture** *capture-name* **stop** **Example:** `Device# monitor capture mycap stop`	Stops the capture of packet data at a traffic trace point.
Step 9	**end** **Example:** `Device# end`	Exits privileged EXEC mode.

Monitoring and Maintaining Captured Data

Perform this task to monitor and maintain the packet data captured. Capture buffer details and capture point details are displayed.

SUMMARY STEPS

1. **enable**
2. **show monitor capture** *capture-buffer-name* **buffer dump**
3. **show monitor capture** *capture-buffer-name* **parameter**
4. **debug epc capture-point**
5. **debug epc provision**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show monitor capture** *capture-buffer-name* **buffer dump** **Example:** `Device# show monitor capture mycap buffer dump`	(Optional) Displays a hexadecimal dump of captured packet and its metadata.
Step 3	**show monitor capture** *capture-buffer-name* **parameter** **Example:** `Device# show monitor capture mycap parameter`	(Optional) Displays a list of commands that were used to specify the capture.
Step 4	**debug epc capture-point** **Example:** `Device# debug epc capture-point`	(Optional) Enables packet capture point debugging.

	Command or Action	Purpose
Step 5	**debug epc provision** **Example:** `Device# debug epc provision`	(Optional) Enables packet capture provisioning debugging.
Step 6	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode.

Configuration Examples for Embedded Packet Capture

Example: Managing Packet Data Capture

The following example shows how to manage packet data capture:

```
Device> enable
Device# monitor capture mycap access-list v4acl
Device# monitor capture mycap limit duration 1000
Device# monitor capture mycap interface GigabitEthernet 0/0/1 both
Device# monitor capture mycap buffer circular size 10
Device# monitor capture mycap start
Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap
Device# monitor capture mycap stop
Device# end
```

Example: Monitoring and Maintaining Captured Data

The following example shows how to dump packets in ASCII format:

```
Device# show monitor capture mycap buffer dump

0
 0000:  01005E00 00020000 0C07AC1D 080045C0   ..^...........E.
 0010:  00300000 00000111 CFDC091D 0002E000   .0..............
 0020:  000207C1 07C1001C 802A0000 10030AFA   .........*......
 0030:  1D006369 73636F00 0000091D 0001       ..example.......

1
 0000:  01005E00 0002001B 2BF69280 080046C0   ..^.....+.....F.
 0010:  00200000 00000102 44170000 0000E000   . ......D.......
 0020:  00019404 00001700 E8FF0000 0000       ..............

2
 0000:  01005E00 0002001B 2BF68680 080045C0   ..^.....+.....E.
 0010:  00300000 00000111 CFDB091D 0003E000   .0..............
 0020:  000207C1 07C1001C 88B50000 08030A6E   ...............n
 0030:  1D006369 73636F00 0000091D 0001       ..example.......

3
```

```
 0000:  01005E00 000A001C 0F2EDC00 080045C0   ..^...........E.
 0010:  003C0000 00000258 CE7F091D 0004E000   .<.....X........
 0020:  000A0205 F3000000 00000000 00000000   ................
 0030:  00000000 00D10001 000C0100 01000000   ................
 0040:  000F0004 00080501 0300                ................
```

The following example shows how to display the list of commands used to configure the capture named mycap:

```
Device# show monitor capture mycap parameter

  monitor capture mycap interface GigabitEthernet 1/0/1 both
  monitor capture mycap match any
  monitor capture mycap buffer size 10
  monitor capture mycap limit pps 1000
```

The following example shows how to debug the capture point:

```
Device# debug epc capture-point

EPC capture point operations debugging is on
Device# monitor capture mycap start

*Jun  4 14:17:15.463: EPC CP:  Starting the capture cap1
*Jun  4 14:17:15.463: EPC CP:  (brief=3, detailed=4, dump=5) = 0
*Jun  4 14:17:15.463: EPC CP:  final check before activation
*Jun  4 14:17:15.463: EPC CP:  setting up c3pl infra
*Jun  4 14:17:15.463: EPC CP:  Setup c3pl acl-class-policy
*Jun  4 14:17:15.463: EPC CP:  Creating a class
*Jun  4 14:17:15.464: EPC CP:  Creating a class : Successful
*Jun  4 14:17:15.464: EPC CP:  class-map Created
*Jun  4 14:17:15.464: EPC CP: creating  policy-name epc_policy_cap1
*Jun  4 14:17:15.464: EPC CP:  Creating Policy epc_policy_cap1 of type 49 and client type
21
*Jun  4 14:17:15.464: EPC CP:  Storing a Policy
*Jun  4 14:17:15.464: EPC CP:  calling ppm_store_policy with epc_policy
*Jun  4 14:17:15.464: EPC CP:  Creating Policy : Successful
*Jun  4 14:17:15.464: EPC CP:  policy-map created
*Jun  4 14:17:15.464: EPC CP:  creating filter for ANY
*Jun  4 14:17:15.464: EPC CP:  Adding acl to class : Successful
*Jun  4 14:17:15.464: EPC CP:  Setup c3pl class to policy
*Jun  4 14:17:15.464: EPC CP:  Attaching Class to Policy
*Jun  4 14:17:15.464: EPC CP:  Attaching epc_class_cap1 to epc_policy_cap1
*Jun  4 14:17:15.464: EPC CP:  Attaching Class to Policy : Successful
*Jun  4 14:17:15.464: EPC CP:  setting up c3pl qos
*Jun  4 14:17:15.464: EPC CP:  DBG> Set packet rate limit to 1000
*Jun  4 14:17:15.464: EPC CP: creating action for policy_map epc_policy_cap1 class_map
epc_class_cap1
*Jun  4 14:17:15.464: EPC CP:  DBG> Set packet rate limit to 1000
*Jun  4 14:17:15.464: EPC CP:  Activating Interface GigabitEthernet1/0/1  direction both
*Jun  4 14:17:15.464: EPC CP: Id attached 0
*Jun  4 14:17:15.464: EPC CP:  inserting into active lists
*Jun  4 14:17:15.464: EPC CP: Id attached 0
*Jun  4 14:17:15.465: EPC CP:  inserting into active lists
*Jun  4 14:17:15.465: EPC CP:  Activating Vlan
*Jun  4 14:17:15.465: EPC CP:  Deleting all temp interfaces
*Jun  4 14:17:15.465: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.
*Jun  4 14:17:15.465: EPC CP:  Active Capture 1

Device# monitor capture mycap1 stop

*Jun  4 14:17:31.963: EPC CP:  Stopping the capture cap1
```

```
*Jun  4 14:17:31.963: EPC CP: Warning: unable to unbind capture cap1
*Jun  4 14:17:31.963: EPC CP:  Deactivating policy-map
*Jun  4 14:17:31.963: EPC CP:  Policy epc_policy_cap1
*Jun  4 14:17:31.964: EPC CP:  Deactivating policy-map Successful
*Jun  4 14:17:31.964: EPC CP:  removing povision feature
*Jun  4 14:17:31.964: EPC CP:  Found action for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun  4 14:17:31.964: EPC CP:  cleanning up c3pl infra
*Jun  4 14:17:31.964: EPC CP:  Removing Class epc_class_cap1 from Policy
*Jun  4 14:17:31.964: EPC CP:  Removing Class from epc_policy_cap1
*Jun  4 14:17:31.964: EPC CP:  Successfully removed
*Jun  4 14:17:31.964: EPC CP:  Removing acl mac from class
*Jun  4 14:17:31.964: EPC CP:  Removing acl from class : Successful
*Jun  4 14:17:31.964: EPC CP:  Removing all policies
*Jun  4 14:17:31.964: EPC CP:  Removing Policy  epc_policy_cap1
*Jun  4 14:17:31.964: EPC CP:  Removing Policy : Successful
*Jun  4 14:17:31.964: EPC CP:  Removing class epc_class_cap1
*Jun  4 14:17:31.965: EPC CP:  Removing class : Successful
*Jun  4 14:17:31.965: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
*Jun  4 14:17:31.965: EPC CP:  Active Capture 0
```

The following example shows how to debug the Embedded Packet Capture (EPC) provisioning:

```
Device# debug epc provision

EPC provisionioning debugging is on

Device# monitor capture mycap start

*Jun  4 14:17:54.991: EPC PROV: No action found for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun  4 14:17:54.991: EPC PROV:
*Jun  4 14:17:54.991: Attempting to install service policy epc_policy_cap1

*Jun  4 14:17:54.992: EPC PROV: Attached service policy to epc idb subblock
*Jun  4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun  4 14:17:54.992: EPC PROV:
*Jun  4 14:17:54.992: Attempting to install service policy epc_policy_cap1

*Jun  4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun  4 14:17:54.992: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.

Device# monitor capture mycap stop

*Jun  4 14:18:02.503: EPC PROV: Successful. Remove feature object
*Jun  4 14:18:02.504: EPC PROV: Successful. Remove feature object
*Jun  4 14:18:02.504: EPC PROV: Destroyed epc idb subblock
*Jun  4 14:18:02.504: EPC PROV: Found action for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun  4 14:18:02.504: EPC PROV: Deleting EPC action
*Jun  4 14:18:02.504: EPC PROV: Successful. CLASS_REMOVE, policy-map epc_policy_cap1, class
 epc_class_cap1
*Jun  4 14:18:02.504: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
```

The following example shows the interfaces that are available during a crypto flexvpn session.

Note If the device supports an LTE module, the cellular interface is displayed under the **show ip interface brief**

```
Device# show ip interface brief

Interface              IP-Address      OK? Method Status                Protocol
Cellular                               YES        up                    up
GigabitEthernet1       9.45.7.78       YES NVRAM  up                    up
GigabitEthernet2       1.1.1.2         YES NVRAM  up                    up
GigabitEthernet3       unassigned      YES NVRAM  administratively down down
GigabitEthernet4       unassigned      YES NVRAM  administratively down down
GigabitEthernet5       unassigned      YES NVRAM  administratively down down
Loopback100            7.1.1.100       YES manual up                    up
Loopback200            8.8.8.100       YES manual up                    up
Loopback300            18.18.18.100    YES manual up                    up
Virtual-Access1        7.1.1.100       YES unset  up                    up
Virtual-Access2        7.1.1.100       YES unset  up                    up
Virtual-Template1      7.1.1.100       YES unset  up                    down
```

Note A virtual-access is available when the interface is dynamically created through a crypto (flexvpn) session using the **monitor capture***capture-name***interface** command.

The following example shows the cellular device operating in the controller mode:

```
Device# show monitor capture EPC

Status Information for Capture EPC
  Target Type: Interface: Cellular0/2/0, Direction: BOTH
Status : Active
  Filter Details: IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 10
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
```

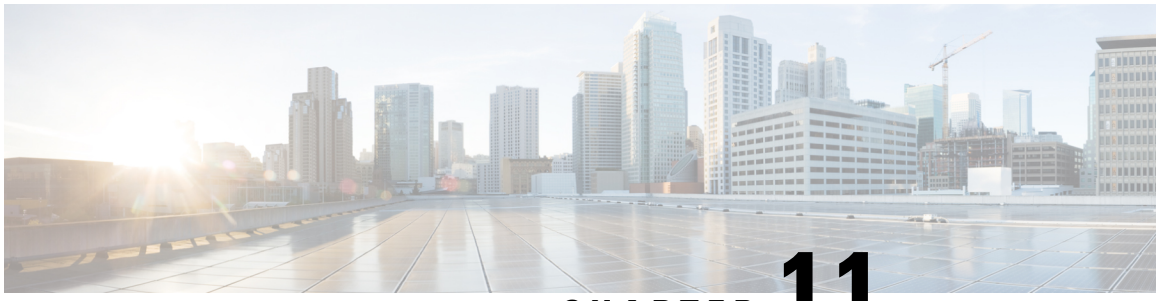
Additional References

Related Documents

Related Topic	Document Title
Embedded Packet Capture commands	Cisco IOS Embedded Packet Capture Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 11

Encrypted Traffic Analytics

Encrypted Traffic Analytics (ET-Analytics) is used to identify malware communications in encrypted traffic. ET-Analytics uses passive monitoring, extraction of relevant data elements, and supervised machine learning with cloud-based global visibility. ET-Analytics exports the relevant data elements in the form of NetFlow record fields to detect whether the packet flow has malware, and these NetFlow record fields include IDP (initial data packet) and SPLT (Sequence of Packet Length and Time).

Feature Information for Encrypted Traffic Analytics

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16: Feature Information for Encrypted Traffic Analytics (ET-Analytics)

Feature Name	Releases	Feature Information
Encrypted Traffic Analytics		Encrypted Traffic Analytics (ET-Analytics) is used to identify malware communications in encrypted traffic. ET-Analytics uses passive monitoring, extraction of relevant data elements, and supervised machine learning with cloud-based global visibility. ET-Analytics exports the relevant data elements in the form of NetFlow record fields to detect whether the packet flow has malware, and these NetFlow record fields include IDP (initial data packet) and SPLT (Sequence of Packet Length and Time).

Restrictions for Encrypted Traffic Analytics

- ET-Analytics is not supported on management interfaces, VRF-Aware Software Infrastructure (VASI) interface, and internal interfaces.
- ET-Analytics is not supported on Switch Port Analyzer (SPAN) ports.

Information About Encrypted Traffic Analytics

Data Elements for Encrypted Traffic

ET-Analytics uses intraflow metadata to identify malware components, maintaining the integrity of the encrypted traffic without the need for bulk decryption and without compromising on data integrity.

ET-Analytics extracts the following main data elements from the network flow: the sequence of packet lengths and times (SPLT), TLS-specific features, and the initial data packet (IDP). Cisco's Application-Specific Integrated Circuit (ASIC) architecture provides the ability to extract these data elements without slowing down the data network. Separate templates can be defined for each of the data elements.

Transport Layer Security (TLS) is a cryptographic protocol that provides privacy for applications. TLS is usually implemented with common protocols such as HTTP for web browsing or Simple Mail Transfer Protocol (SMTP) for email. HTTPS is the use of TLS over HTTP; this protocol is used to secure communication between a web server and client and is supported by most major web servers.

The TLS template is used to report several of the TLS parameters in use for a flow. These parameters help in finding the use of insecure cipher suites, out-of-date protocol version, and so on.

- Sequence of Packet Lengths and Times (SPLT) - SPLT contains the length (number of bytes) of each packet's application payload for the first several packets of a flow, along with the inter-arrival times of those packets. SPLT can be represented as an array of packet sizes (in bytes) along with an array of times (in milliseconds) indicating the time since the previous packet was observed. The SPLT template is used to report packet size and timing information for a flow, which is useful to analyze encrypted traffic and find malicious flows or perform other classifications.
- Initial Data Packet (IDP) - IDP obtains packet data from the first packet of a flow. It allows extraction of data such as an HTTP URL, DNS hostname/address, and other data elements. The TLS handshake is composed of several messages that contain unencrypted metadata used to extract data elements such as cipher suites, TLS versions, and the client's public key length. The IDP template is used to report packet data from the first data packet of a flow. This template allows collectors to perform application classification of a flow (for example, using Snort).

How to Configure Encrypted Traffic Analytics

Enabling ET-Analytics on an Interface

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**et-analytics**	Enters encrypted traffic analytics configuration mode.
Step 4	**exit**	Returns to global configuration mode.
Step 5	**interface** *interface-id*	Specifies the interface and port number and enters interface configuration mode.
Step 6	**et-analytics enable**	Enables encrypted traffic analytics on this interface.
Step 7	**end**	Returns to privileged EXEC mode.

Example

```
Device> enable
Device# configure terminal
Device(config)# et-analytics
Device(config-et-analytics)# ip flow-export destination 192.0.2.1 2055 vrf green
Device(config-et-analytics)# exit
Device(config)# interface gigabitethernet 0/0/1
Device(config-if)# et-analytics enable
Device(config-if)# end
```

Applying an ACL for Allowed listing

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**et-analytics**	Enters encrypted traffic analytics configuration mode.

	Command or Action	Purpose
Step 4	**whitelist acl** *access-list*	Allowed lists the specified access list traffic. The access list can be a standard, extended, or named ACL.
Step 5	**exit**	Returns to global configuration mode.
Step 6	**ip access-list extended** *access-list*	Specifies a named extended access list and enters extended access list configuration mode.
Step 7	**permit ip** {*ip-address* | **any** | **host** | **object-group**}	Specifies the packets to forward to a source host or source IP address.
Step 8	**end**	Returns to privileged EXEC mode.

Example

```
Device> enable
Device# configure terminal
Device(config)# et-analytics
Device(config-et-analytics)# whitelist acl eta_whitelist
Device(config-et-analytics)# exit
Device(config)# ip access-list extended eta_whitelist
Device(config-ext-nacl)# permit ip host 198.51.100.1 any
Device(config-ext-nacl)# permit ip any host 198.51.100.1
Device(config-ext-nacl)# permit ip host 198.51.200.1 any
Device(config-ext-nacl)# permit ip any host 198.51.200.1
Device(config-ext-nacl)# end
```

Verifying the ET-Analytics Configuration

Use the following commands for releases earlier to Cisco IOS XE Gibraltar 16.11.1. The following **show** commands are used to see the platform ET-analytics, threat-visibility interfaces, FMAN FP global and interface information, and ET-analytics datapath information. Given below are the sample outputs of the **show** commands.

```
Device# show platform hardware qfp active feature et-analytics data interface gigabitEthernet
 2

uidb handle: 0x3fe
Interface Name: GigabitEthernet2
```

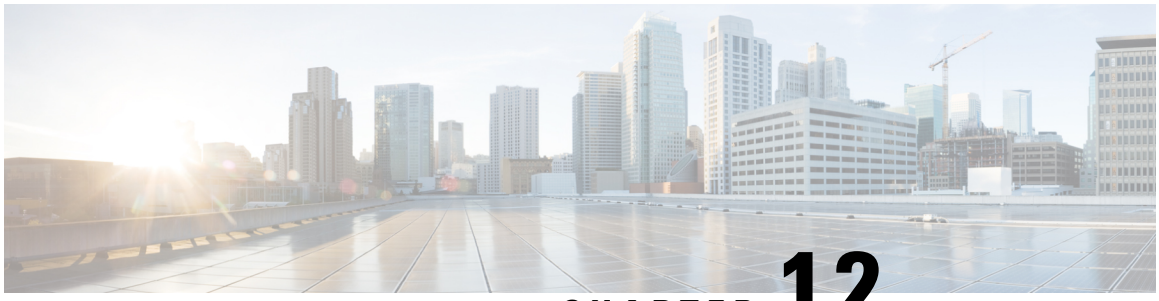
```
Device# show platform hardware qfp active feature et-analytics data memory

  ET-Analytics memory information:

   Size of FO            : 3200 bytes
   No. of FO allocs      : 952903
   No. of FO frees       : 952902
```

```
Device# show platform hardware qfp active feature et-analytics data runtime

  ET-Analytics run-time information:

   Feature state         : initialized (0x00000004)
   Inactive timeout      : 15 secs (default 15 secs)
   Flow CFG information  :   !Flow Table Infrastructure information internal to ETA!
       instance ID       : 0x0
       feature ID        : 0x0
       feature object ID : 0x0
       chunk ID          : 0x4
```

```
Device# show platform hardware qfp active feature et-analytics datapath stats export

  ET-Analytics 192.168.1.100:2055 vrf 2 Stats:
    Export statistics:
      Total records exported     : 2967386
      Total packets exported     : 1885447
      Total bytes exported       : 2056906120
      Total dropped records      : 0
      Total dropped packets      : 0
      Total dropped bytes        : 0
      Total IDP records exported :
            initiator->responder : 805813
            responder->initiator : 418799
      Total SPLT records exported:
            initiator->responder : 805813
            responder->initiator : 418799
      Total SALT records exported:
            initiator->responder : 0
            responder->initiator : 0
      Total BD records exported  :
            initiator->responder : 0
            responder->initiator : 0
      Total TLS records exported :
            initiator->responder : 171332
            responder->initiator : 174860
ET-Analytics 172.27.56.99:2055 Stats:
    Export statistics:
      Total records exported     : 2967446
      Total packets exported     : 1885448
      Total bytes exported       : 2056909280
      Total dropped records      : 0
      Total dropped packets      : 0
      Total dropped bytes        : 0
      Total IDP records exported :
            initiator->responder : 805813
            responder->initiator : 418799
      Total SPLT records exported:
            initiator->responder : 805813
            responder->initiator : 418799
   Total SALT records exported:
            initiator->responder : 0
            responder->initiator : 0
      Total BD records exported  :
            initiator->responder : 0
            responder->initiator : 0
      Total TLS records exported :
            initiator->responder : 171332
            responder->initiator : 174860
```

```
Device# show platform hardware qfp active feature et-analytics datapath stats flow

  ET-Analytics Stats:
    Flow statistics:
      feature object allocs : 0
      feature object frees  : 0
      flow create requests  : 0
      flow create matching  : 0
      flow create successful: 0
      flow create failed, CFT handle: 0
      flow create failed, getting FO: 0
      flow create failed, malloc FO : 0
      flow create failed, attach FO : 0
      flow create failed, match flow: 0
      flow create, aging already set: 0
      flow ageout requests          : 0
      flow ageout failed, freeing FO: 0
      flow ipv4 ageout requests     : 0
      flow ipv6 ageout requests     : 0
      flow whitelist traffic match  : 0
```

```
Device# show vrf tableid

VRF Name                            Tableid             Address Family
Mgmt-intf                           0x00000001          ipv4 unicast
Mgmt-intf                           0x1E000001          ipv6 unicast
blu                                 0x00000002          ipv4 unicast
red                                 0x00000003          ipv4 unicast
```



CHAPTER 12

Flexible Netflow Overview

NetFlow is a Cisco IOS technology that provides statistics on packets flowing through the router. NetFlow is the standard for acquiring IP operational data from IP networks. NetFlow provides data to enable network and security monitoring, network planning, traffic analysis, and IP accounting.

Flexible NetFlow improves on original NetFlow by adding the capability to customize the traffic analysis parameters for your specific requirements. Flexible NetFlow facilitates the creation of more complex configurations for traffic analysis and data export through the use of reusable configuration components.

Prerequisites for Flexible NetFlow

- You are familiar with the Flexible NetFlow key fields as they are defined in the following commands:
  - **match flow**
  - **match interface**
  - **match** {**ipv4** | **ipv6**}
  - **match routing**
  - **match transport**
- You are familiar with the Flexible NetFlow nonkey fields as they are defined in the following commands:
  - **collect counter**
  - **collect flow**
  - **collect interface**
  - **collect**{**ipv4** | **ipv6**}
  - **collect routing**
  - **collect timestamp sys-uptime**
  - **collect transport**

- The networking device must be running a Cisco release that supports Flexible NetFlow.

IPv4 Traffic

- The networking device must be configured for IPv4 routing.
- One of the following must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding or distributed Cisco Express Forwarding.

IPv6 Traffic

- The networking device must be configured for IPv6 routing.
- One of the following must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding IPv6 or distributed Cisco Express Forwarding.

Restrictions for Flexible Netflow

- It is recommended that the total dataplane memory consumed by Flexible Netflow or Original Netflow is limited to a maximum of 25% of the amount of data plane DRAM for an ESP/FP.
- Flexible Netflow export will not work over an IPSEC VPN tunnel if the source of the netflow data is the same router where the VPN tunnel is terminated unless you configure the output-features command under the flow exporter.

Information About Flexible Netflow

Flexible NetFlow Overview

Flexible NetFlow facilitates the creation of more complex configurations for traffic analysis and data export through the use of reusable configuration components.

Typical Uses for NetFlow

NetFlow is typically used for several key customer applications, including the following:

- Network monitoring. NetFlow data enables extensive near-real-time network monitoring capabilities. Flow-based analysis techniques are used by network operators to visualize traffic patterns associated with individual routers and switches and network-wide traffic patterns (providing aggregate traffic or application-based views) to provide proactive problem detection, efficient troubleshooting, and rapid problem resolution.
- Application monitoring and profiling. NetFlow data enables network managers to gain a detailed time-based view of application usage over the network. This information is used to plan, understand new services, and allocate network and application resources (for example, web server sizing and VoIP deployment) to meet customer demands responsively.

- User monitoring and profiling. NetFlow data enables network engineers to gain detailed understanding of customer and user use of network and application resources. This information may then be used to efficiently plan and allocate access, backbone, and application resources and to detect and resolve potential security and policy violations.
- Network planning. NetFlow can be used to capture data over a long period of time, affording the opportunity to track and anticipate network growth and plan upgrades to increase the number of routing devices, ports, and higher-bandwidth interfaces. NetFlow services data optimizes network planning for peering, backbone upgrades, and routing policy. NetFlow helps to minimize the total cost of network operations while maximizing network performance, capacity, and reliability. NetFlow detects unwanted WAN traffic, validates bandwidth and quality of service (QoS), and allows the analysis of new network applications. NetFlow will give you valuable information to reduce the cost of operating your network.
- Security analysis. NetFlow identifies and classifies distributed denial of service (dDoS) attacks, viruses, and worms in real time. Changes in network behavior indicate anomalies that are clearly demonstrated in Flexible NetFlow data. The data is also a valuable forensic tool to understand and replay the history of security incidents.
- Billing and accounting. NetFlow data provides fine-grained metering (for instance, flow data includes details such as IP addresses, packet and byte counts, time stamps, type of service (ToS), and application ports) for highly flexible and detailed resource utilization accounting. Service providers may use the information for billing based on time of day, bandwidth usage, application usage, quality of service, and so on. Enterprise customers may use the information for departmental charge back or cost allocation for resource utilization.
- NetFlow data warehousing and data mining. NetFlow data (or derived information) can be warehoused for later retrieval and analysis in support of proactive marketing and customer service programs (for example, discovering which applications and services are being used by internal and external users and targeting them for improved service, advertising, and so on). In addition, Flexible NetFlow data gives market researchers access to the "who," "what," "where," and "how long" information relevant to enterprises and service providers.

Use of Flows in Original NetFlow and Flexible NetFlow

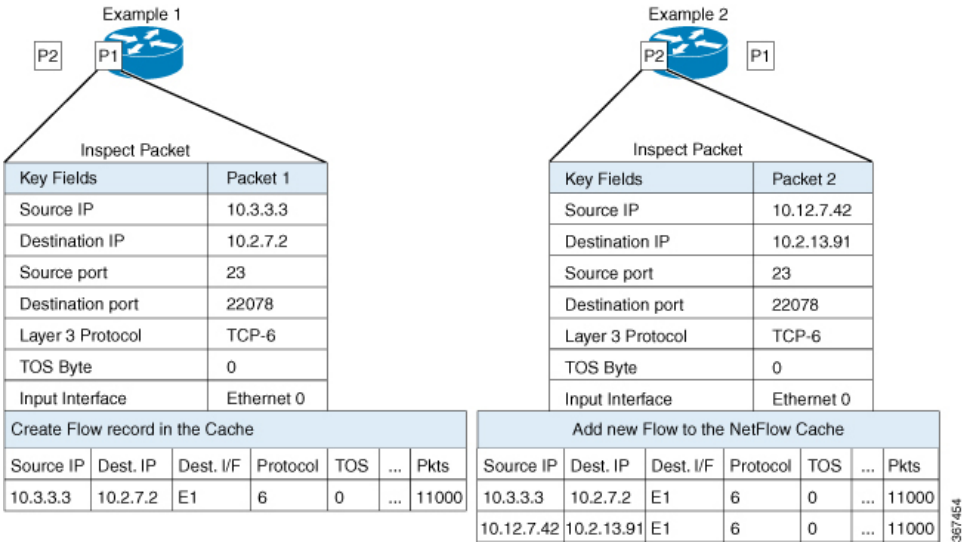
Original NetFlow and Flexible NetFlow both use the concept of flows. A *flow* is defined as a stream of packets between a given source and a given destination.

Original NetFlow and Flexible NetFlow both use the values in key fields in IP datagrams, such as the IP source or destination address and the source or destination transport protocol port, as the criteria for determining when a new flow must be created in the cache while network traffic is being monitored. When the value of the data in the key field of a datagram is unique with respect to the flows that already exist, a new flow is created.

Original NetFlow and Flexible NetFlow both use nonkey fields as the criteria for identifying fields from which data is captured from the flows. The flows are populated with data that is captured from the values in the nonkey fields.

The figure below is an example of the process for inspecting packets and creating flow records in the cache. In this example, two unique flows are created in the cache because different values are in the source and destination IP address key fields.

Figure 2: Packet Inspection



Original NetFlow and Benefits of Flexible NetFlow

Flexible NetFlow allows the flow to be user defined. The benefits of Flexible NetFlow include:

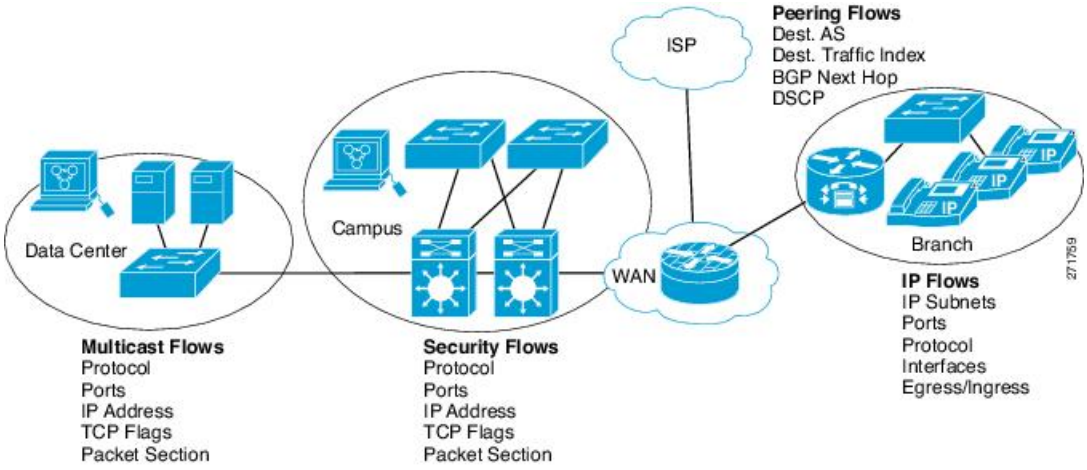
- High-capacity flow recognition, including scalability and aggregation of flow information.
- Enhanced flow infrastructure for security monitoring and dDoS detection and identification.
- New information from packets to adapt flow information to a particular service or operation in the network. The flow information available will be customizable by Flexible NetFlow users.
- Extensive use of Cisco's flexible and extensible NetFlow Version 9.
- A comprehensive IP accounting feature that can be used to replace many accounting features, such as IP accounting, Border Gateway Protocol (BGP) Policy Accounting, and persistent caches.

Flexible NetFlow allows you to understand network behavior with more efficiency, with specific flow information tailored for various services used in the network. The following are some example applications for a Flexible NetFlow feature:

- Flexible NetFlow enhances Cisco NetFlow as a security monitoring tool. For instance, new flow keys can be defined for packet length or MAC address, allowing users to search for a specific type of attack in the network.
- Flexible NetFlow allows you to quickly identify how much application traffic is being sent between hosts by specifically tracking TCP or UDP applications by the class of service (CoS) in the packets.
- The accounting of traffic entering a Multiprotocol Label Switching (MPLS) or IP core network and its destination for each next hop per class of service. This capability allows the building of an edge-to-edge traffic matrix.

The figure below is an example of how Flexible NetFlow might be deployed in a network.

Figure 3: Typical Deployment for Flexible NetFlow



Flexible NetFlow Components

Flexible NetFlow consists of components that can be used together in several variations to perform traffic analysis and data export. The user-defined flow records and the component structure of Flexible NetFlow facilitates the creation of various configurations for traffic analysis and data export on a networking device with a minimum number of configuration commands. Each flow monitor can have a unique combination of flow record, flow exporter, and cache type. If you change a parameter such as the destination IP address for a flow exporter, it is automatically changed for all the flow monitors that use the flow exporter. The same flow monitor can be used in conjunction with different flow samplers to sample the same type of network traffic at different rates on different interfaces. The following sections provide more information on Flexible NetFlow components:

Flow Records

In Flexible NetFlow a combination of key and non-key fields is called a *flow record*. Flexible NetFlow records are assigned to Flexible NetFlow flow monitors to define the cache that is used for storing flow data. Flexible NetFlow includes several predefined records that can help you get started using Flexible NetFlow.

To use Flexible NetFlow to its fullest potential, you need to create your own customized records, as described in the following section(s):

NetFlow Predefined Records

Flexible NetFlow includes several predefined records that you can use to start monitoring traffic in your network. The predefined records are available to help you quickly deploy Flexible NetFlow and are easier to use than user-defined flow records. You can choose from a list of already defined records that may meet the needs for network monitoring. As Flexible NetFlow evolves, popular user-defined flow records will be made available as predefined records to make them easier to implement.

The predefined records ensure backward compatibility with your existing NetFlow collector configurations for the data that is exported. Each of the predefined records has a unique combination of key and nonkey fields that offer you the built-in ability to monitor various types of traffic in your network without customizing Flexible NetFlow on your router.

Two of the predefined records (NetFlow original and NetFlow IPv4/IPv6 original output), which are functionally equivalent, emulate original (ingress) NetFlow and the Egress NetFlow Accounting feature in original NetFlow, respectively. Some of the other Flexible NetFlow predefined records are based on the aggregation cache schemes available in original NetFlow. The Flexible NetFlow predefined records that are based on the aggregation cache schemes available in original NetFlow do not perform aggregation. Instead each flow is tracked separately by the predefined records.

User-Defined Records

Flexible NetFlow enables you to define your own records for a Flexible NetFlow flow monitor cache by specifying the key and nonkey fields to customize the data collection to your specific requirements. When you define your own records for a Flexible NetFlow flow monitor cache, they are referred to as *user-defined records*. The values in nonkey fields are added to flows to provide additional information about the traffic in the flows. A change in the value of a nonkey field does not create a new flow. In most cases the values for nonkey fields are taken from only the first packet in the flow. Flexible NetFlow enables you to capture counter values such as the number of bytes and packets in a flow as nonkey fields.

Flexible NetFlow adds a new Version 9 export format field type for the header and packet section types. Flexible NetFlow will communicate to the NetFlow collector the configured section sizes in the corresponding Version 9 export template fields. The payload sections will have a corresponding length field that can be used to collect the actual size of the collected section.

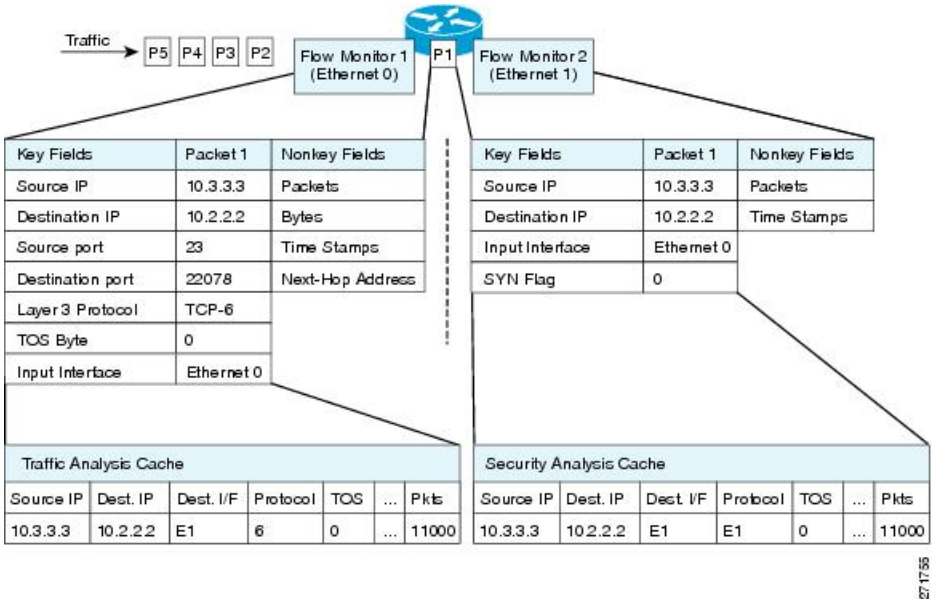
Flow Monitors

Flow monitors are the Flexible NetFlow component that is applied to interfaces to perform network traffic monitoring.

Flow data is collected from the network traffic and added to the flow monitor cache during the monitoring process based on the key and nonkey fields in the flow record.

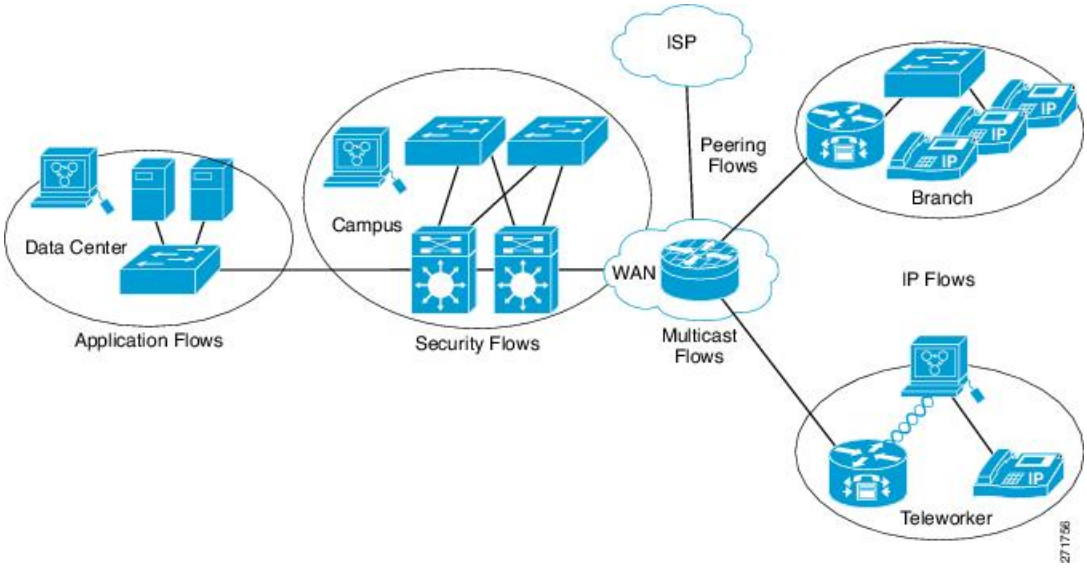
Flexible NetFlow can be used to perform different types of analysis on the same traffic. In the figure below, packet 1 is analyzed using a record designed for standard traffic analysis on the input interface and a record designed for security analysis on the output interface.

Figure 4: Example of Using Two Flow Monitors to Analyze the Same Traffic



The figure below shows a more complex example of how you can apply different types of flow monitors with custom records.

Figure 5: Complex Example of Using Multiple Types of Flow Monitors with Custom Records



Normal

The default cache type is “normal”. In this mode, the entries in the cache are aged out according to the timeout active and timeout inactive settings. When a cache entry is aged out, it is removed from the cache and exported via any exporters configured.

Flow Exporters

Flow exporters export the data in the flow monitor cache to a remote system, such as a server running NetFlow collector, for analysis and storage. Flow exporters are created as separate entities in the configuration. Flow exporters are assigned to flow monitors to provide data export capability for the flow monitors. You can create several flow exporters and assign them to one or more flow monitors to provide several export destinations. You can create one flow exporter and apply it to several flow monitors.

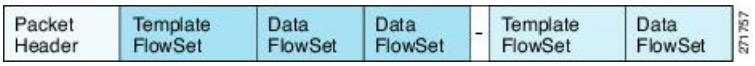
NetFlow Data Export Format Version 9

The basic output of NetFlow is a flow record. Several different formats for flow records have evolved as NetFlow has matured. The most recent evolution of the NetFlow export format is known as Version 9. The distinguishing feature of the NetFlow Version 9 export format is that it is template-based. Templates provide an extensible design to the record format, a feature that should allow future enhancements to NetFlow services without requiring concurrent changes to the basic flow-record format. Using templates provides several key benefits:

- Third-party business partners who produce applications that provide collector or display services for NetFlow do not have to recompile their applications each time a new NetFlow feature is added. Instead, they should be able to use an external data file that documents the known template formats.
- New features can be added to NetFlow quickly without breaking current implementations.
- NetFlow is “future-proofed” against new or developing protocols because the Version 9 format can be adapted to provide support for them.

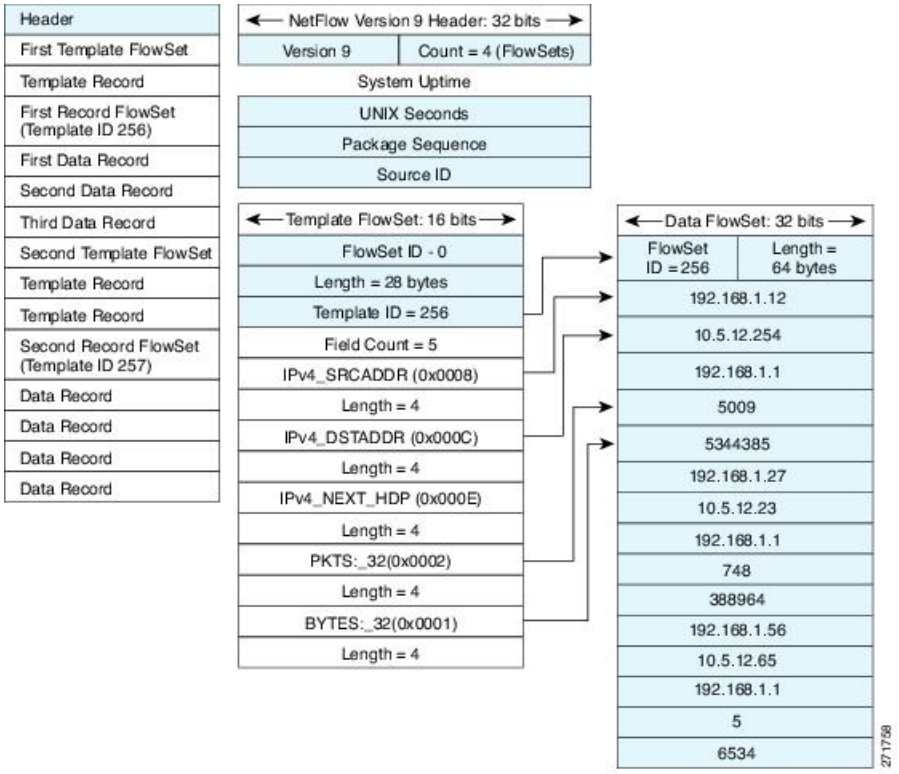
The Version 9 export format consists of a packet header followed by one or more template flow or data flow sets. A template flow set provides a description of the fields that will be present in future data flow sets. These data flow sets may occur later within the same export packet or in subsequent export packets. Template flow and data flow sets can be intermingled within a single export packet, as illustrated in the figure below.

Figure 6: Version 9 Export Packet



NetFlow Version 9 will periodically export the template data so the NetFlow collector will understand what data is to be sent and also export the data flow set for the template. The key advantage to Flexible NetFlow is that the user configures a flow record, which is effectively converted to a Version 9 template and then forwarded to the collector. The figure below is a detailed example of the NetFlow Version 9 export format, including the header, template flow, and data flow sets.

Figure 7: Detailed Example of the NetFlow Version 9 Export Format



Flow Samplers

Flow samplers are created as separate components in a router's configuration. Flow samplers are used to reduce the load on the device that is running by limiting the number of packets that are selected for analysis.

Flow sampling exchanges monitoring accuracy for router performance. When you apply a sampler to a flow monitor, the overhead load on the router of running the flow monitor is reduced because the number of packets that the flow monitor must analyze is reduced. The reduction in the number of packets that are analyzed by the flow monitor causes a corresponding reduction in the accuracy of the information stored in the flow monitor's cache.

Samplers are combined with flow monitors when they are applied to an interface with the **ip flow monitor** command.

Security Monitoring with Flexible NetFlow

Flexible NetFlow can be used as a network attack detection tool with capabilities to track all parts of the IP header and even packet sections and characterize this information into flows. Security monitoring systems can analyze Flexible NetFlow data, and upon finding an issue in the network, create a virtual bucket or virtual cache that will be configured to track specific information and identify details about the attack pattern or worm propagation. The capability to create caches dynamically with specific information combined with input filtering (for example, filtering all flows to a specific destination) makes Flexible NetFlow a powerful security monitoring tool.

One common type of attack occurs when TCP flags are used to flood open TCP requests to a destination server (for example, a SYN flood attack). The attacking device sends a stream of TCP SYNs to a given destination address but never sends the ACK in response to the servers SYN-ACK as part of the TCP three-way handshake. The flow information needed for a security detection server requires the tracking of three key fields: destination address or subnet, TCP flags, and packet count. The security detection server may be monitoring general Flexible NetFlow information, and this data may trigger a detailed view of this particular attack by the Flexible NetFlow dynamically creating a new flow monitor in the router's configuration. The new flow monitor might include input filtering to limit what traffic is visible in the Flexible NetFlow cache along with the tracking of the specific information to diagnose the TCP-based attack. In this case the user may want to filter all flow information to the server destination address or subnet to limit the amount of information the security detection server needs to evaluate. If the security detection server decided it understood this attack, it might then program another flow monitor to collect and export payload information or sections of packets to take a deeper look at a signature within the packet. This example is just one of many possible ways that Flexible NetFlow can be used to detect security incidents.

Feature Comparison of Original NetFlow and Flexible NetFlow

The table below provides a feature-by-feature comparison of original NetFlow and Flexible NetFlow.

Table 17: Feature-by-Feature Comparison of Original NetFlow and Flexible NetFlow

Feature	Original NetFlow	Flexible NetFlow	Comments
NetFlow Data Capture	Supported	Supported	Data capture is available with the predefined and user-defined records in Flexible NetFlow. Flexible NetFlow has several predefined keys that emulate the traffic analysis capabilities of original NetFlow.
NetFlow Data Export	Supported	Supported	Flow exporters export data from the Flexible NetFlow flow monitor caches to remote systems.
NetFlow for IPv6	Supported	Supported	IPv6 support was removed from original NetFlow in Cisco IOS Release 12.4(20)T. The Flexible NetFlow--IPv6 Unicast Flows feature implemented IPv6 support for Flexible NetFlow in Cisco IOS Release 12.4(20)T.
NetFlow BGP Next Hop Support	Supported	Supported	Available in the predefined and user-defined keys in Flexible NetFlow records.
Random Packet Sampled NetFlow	Supported	Supported	Available with Flexible NetFlow sampling.
NetFlow v9 Export Format	Supported	Supported	Available with Flexible NetFlow exporters.
NetFlow Subinterface Support	Supported	Supported	Flexible NetFlow monitors can be assigned to subinterfaces.

Feature	Original NetFlow	Flexible NetFlow	Comments
NetFlow Multiple Export Destinations	Supported	Supported	Available with Flexible NetFlow exporters.
NetFlow ToS-Based Router Aggregation	Supported	Supported	Available in the predefined and user-defined records in Flexible NetFlow records.
NetFlow Minimum Prefix Mask for Router-Based Aggregation	Supported	Supported	Available in the predefined and user-defined records.
NetFlow Input Filters	Supported	Not supported	--
NetFlow MIB	Supported	Not supported	--
Egress NetFlow Accounting	Supported	Supported	Flexible NetFlow monitors can be used to monitor egress traffic on interfaces and subinterfaces.

Criteria for Identifying Traffic to Be Used in Analysis in Flexible NetFlow

If the predefined Flexible NetFlow records are not suitable for your traffic requirements, you can create a user-defined (custom) record using the Flexible NetFlow **collect** and **match** commands. Before you can create a customized record, you must decide the criteria that you are going to use for the key and nonkey fields.

If you want to create a customized record for detecting network attacks, you must include the appropriate key and nonkey fields in the record to ensure that the router creates the flows and captures the data that you need to analyze the attack and respond to it. For example, SYN flood attacks are a common denial of service (DoS) attack in which TCP flags are used to flood open TCP requests to a destination host. When a normal TCP connection starts, a destination host receives a SYN (synchronize/start) packet from a source host and sends back a SYN ACK (synchronize acknowledge). The destination host must then hear an ACK (acknowledge) of the SYN ACK before the connection is established. This is referred to as the "TCP three-way handshake While the destination host waits for the ACK to the SYN ACK, a connection queue of finite size on the destination host keeps track of connections waiting to be completed. This queue typically empties quickly because the ACK is expected to arrive a few milliseconds after the SYN ACK. The TCP SYN attack exploits this design by having an attacking source host generate TCP SYN packets with random source addresses toward a victim host. The victim destination host sends a SYN ACK back to the random source address and adds an entry to the connection queue. Because the SYN ACK is destined for an incorrect or nonexistent host, the last part of the TCP three-way handshake is never completed and the entry remains in the connection queue until a timer expires, typically for about one minute. When the source host rapidly generates TCP SYN packets from random IP addresses, the connection queue can be filled and TCP services (such as e-mail, file transfer, or WWW) can be denied to legitimate users.

The information needed for a security monitoring record for this type of DoS attack might include the following key and nonkey fields:

- Key fields:
  - Destination IP address or destination IP subnet
  - TCP flags
  - Packet count

- Nonkey fields
  - Destination IP address
  - Source IP address
  - Interface input and output

Tip Many users configure a general Flexible NetFlow monitor that triggers a more detailed Flexible NetFlow view of a DoS attack using these key and nonkey fields.

Benefit of Emulating Original NetFlow with Flexible NetFlow

Emulating original NetFlow with Flexible NetFlow enables to you to deploy Flexible NetFlow quickly because you can use a predefined record instead of designing and configuring a custom user-defined record. You need only configure a flow monitor and apply it to an interface for Flexible NetFlow to start working like original NetFlow. You can add an optional exporter if you want to analyze the data that you collect with an application such as NetFlow collector.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. The record format can be one of the predefined record formats, or an advanced user may create his or her own record format using the **collect** and **match**commands in Flexible NetFlow flow record configuration mode.

Flow exporters are used to send the data that you collect with Flexible NetFlow to a remote system such as a NetFlow Collection Engine. Exporters use UDP as the transport protocol and use the Version 9 export format.

If you are familiar with original NetFlow, you already understand the format and content of the data that you collect and export with Flexible NetFlow when you emulate original NetFlow. You will be able to use the same techniques for analyzing the data.

Flexible NetFlow Predefined Records

Flexible NetFlow predefined records are based on the original NetFlow ingress and egress caches and the aggregation caches. The difference between the original NetFlow aggregation caches and the corresponding predefined Flexible NetFlow records is that the predefined records do not perform aggregation. Flexible NetFlow predefined records are associated with a Flexible NetFlow flow monitor the same way that you associate a user-defined (custom) record.

Benefits of Flexible NetFlow Predefined Records

If you have been using original NetFlow or original NetFlow with aggregation caches you can continue to capture the same traffic data for analysis when you migrate to Flexible NetFlow by using the predefined records available with Flexible NetFlow. Many users will find that the preexisting Flexible NetFlow records are suitable for the majority of their traffic analysis requirements.

NetFlow Original and NetFlow IPv4 Original Input Predefined Records

The Flexible NetFlow "NetFlow original" and "NetFlow IPv4 original input" predefined records can be used interchangeably because they have the same key and nonkey fields. The key and nonkey fields and the counters for the Flexible NetFlow "NetFlow original" and "NetFlow IPv4 original input" predefined records are shown in the table below.

Table 18: Key and Nonkey Fields Used by the Flexible NetFlow NetFlow Original and NetFlow IPv4 Original Input Predefined Records

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the type of service (ToS) field.
IP Protocol	Key	Value in the IP protocol field.
IP Source Address	Key	IP source address.
IP Destination Address	Key	IP destination address.
Transport Source Port	Key	Value of the transport layer source port field.
Transport Destination Port	Key	Value of the transport layer destination port field.
Interface Input	Key	Interface on which the traffic is received.
Flow Sampler ID	Key	ID number of the flow sampler (if flow sampling is enabled).
IP Source AS	Nonkey	Source autonomous system number.
IP Destination AS	Nonkey	Destination autonomous system number.
IP Next Hop Address	Nonkey	IP address of the next hop.
IP Source Mask	Nonkey	Mask for the IP source address.
IP Destination Mask	Nonkey	Mask for the IP destination address.
TCP Flags	Nonkey	Value in the TCP flag field.
Interface Output	Nonkey	Interface on which the traffic is transmitted.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

NetFlow IPv4 Original Output Predefined Record

The Flexible NetFlow "NetFlow IPv4 original output" predefined record is used to emulate the original NetFlow Egress NetFlow Accounting feature that was released in Cisco IOS Release 12.3(11)T. The key and nonkey fields and the counters for the Flexible NetFlow "NetFlow IPv4 original output" predefined record are shown in the table below.

Table 19: Key and Nonkey Fields Used by the Flexible NetFlow NetFlow IPv4 Original Output Predefined Record

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the ToS field.
IP Protocol	Key	Value in the IP protocol field.
IP Source Address	Key	IP source address.
IP Destination Address	Key	IP destination address.
Transport Source Port	Key	Value of the transport layer source port field.
Transport Destination Port	Key	Value of the transport layer destination port field.
Interface Output	Key	Interface on which the traffic is transmitted.
Flow Sampler ID	Key	ID number of the flow sampler (if flow sampling is enabled).
IP Source AS	Nonkey	Source autonomous system number.
IP Destination AS	Nonkey	Destination autonomous system number.
IP Next Hop Address	Nonkey	IP address of the next hop.
IP Source Mask	Nonkey	Mask for the IP source address.
IP Destination Mask	Nonkey	Mask for the IP destination address.
TCP Flags	Nonkey	Value in the TCP flag field.
Interface Input	Nonkey	Interface on which the traffic is received.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

NetFlow IPv6 Original Input Predefined Record

The key and nonkey fields and the counters for the Flexible NetFlow "NetFlow IPv6 original input" predefined record are shown in the table below.

Table 20: Key and Nonkey Fields Used by the Flexible NetFlow NetFlow IPv6 Original Input Predefined Record

Field	Key or NonKey Field	Definition
Traffic Class	Key	Value in the traffic class field.
Flow Label	Key	Flow label.
Protocol	Key	Value in the protocol field.
Extension Map	Key	Value in the extension map bitmap.
IP Source Address	Key	IP source address.
IP Destination Address	Key	IP destination address.
Transport Source Port	Key	Value of the transport layer source port field.
Transport Destination Port	Key	Value of the transport layer destination port field.
Interface Input	Key	Interface on which the traffic is received.
Flow Direction	Key	The direction of the flow.
Flow Sampler	Key	ID number of the flow sampler (if flow sampling is enabled).
Routing Source AS	Nonkey	Source autonomous system number.
Routing Destination AS	Nonkey	Destination autonomous system number.
Routing Next-hop Address	Nonkey	IP address of the next hop.
IP Source Mask	Nonkey	Mask for the IP source address.
IP Destination Mask	Nonkey	Mask for the IP destination address.
Transport TCP Flags	Nonkey	Value in the TCP flag field.
Interface Output	Nonkey	Interface over which the traffic is transmitted.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

NetFlow IPv6 Original Output Predefined Record

The key and nonkey fields and the counters for the Flexible NetFlow "NetFlow IPv6 original output" predefined record are shown in the table below.

Table 21: Key and Nonkey Fields Used by the Flexible NetFlow NetFlow IPv6 Original Output Predefined Record

Field	Key or Nonkey Field	Definition
Traffic Class	Key	Value in the traffic class field.
Flow Label	Key	The flow label.
Protocol	Key	Value in the protocol field.
Extension Map	Key	Value in the extension map bitmap.
IP Source Address	Key	IP source address.
IP Destination Address	Key	IP destination address.
Transport Source Port	Key	Value of the transport layer source port field.
Transport Destination Port	Key	Value of the transport layer destination port field.
Interface Output	Key	Interface over which the traffic is transmitted.
Flow Direction	Key	The direction of the flow.
Flow Sampler	Key	ID number of the flow sampler (if flow sampling is enabled).
Routing Source AS	Nonkey	Source autonomous system number.
Routing Destination AS	Nonkey	Destination autonomous system number.
Routing Next-hop Address	Nonkey	IP address of the next hop.
IP Source Mask	Nonkey	Mask for the IP source address.
IP Destination Mask	Nonkey	Mask for the IP destination address.
Transport TCP Flags	Nonkey	Value in the TCP flag field.
Interface Input	Nonkey	Interface on which the traffic is received.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.

Field	Key or Nonkey Field	Definition
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Autonomous System Predefined Record

The Flexible NetFlow "autonomous system" predefined record creates flows based on autonomous system-to-autonomous system traffic flow data. The Flexible NetFlow "autonomous system" predefined record uses the same key and nonkey fields as the original NetFlow "autonomous system" aggregation cache.

Note This predefined record can be used to analyze IPv4 and IPv6 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "autonomous system" predefined record.

Table 22: Key and Nonkey Fields Used by the Flexible NetFlow Autonomous System Predefined Record

Field	Key or Nonkey Field	Definition
IP Source AS	Key	Autonomous system of the source IP address (peer or origin).
IP Destination AS	Key	Autonomous system of the destination IP address (peer or origin).
Interface Input	Key	Interface on which the traffic is received.
Interface Output	Key	Interface on which the traffic is transmitted.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds since this device was first booted) when the last packet was switched.

Autonomous System ToS Predefined Record

The Flexible NetFlow "autonomous system ToS" predefined record creates flows based on autonomous system-to-autonomous system and type of service (ToS) traffic flow data. The Flexible NetFlow "autonomous system ToS" predefined record uses the same key and nonkey fields as the original NetFlow "autonomous system ToS" aggregation cache.

Note This predefined record can be used to analyze only IPv4 traffic.

Tip This predefined record is particularly useful for generating autonomous system-to-autonomous system traffic flow data.

The table below lists the key and nonkey fields used in the Flexible NetFlow "autonomous system ToS" predefined record.

Table 23: Key and Nonkey Fields Used by the Flexible NetFlow Autonomous System ToS Predefined Record

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the ToS field.
IP Source autonomous system	Key	Autonomous system of the source IP address (peer or origin).
IP Destination autonomous system	Key	Autonomous system of the destination IP address (peer or origin).
Interface Input	Key	Interface on which the traffic is received.
Interface Output	Key	Interface on which the traffic is transmitted.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

BGP Next-Hop Predefined Record

The Flexible NetFlow "BGP next-hop" predefined record creates flows based on Border Gateway Protocol (BGP) traffic flow data.

Note This predefined record can be used to analyze only IPv6 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "BGP next-hop" predefined record.

Table 24: Key and Nonkey Fields Used by the Flexible NetFlow BGP Next-Hop Predefined Record

Field	Key or Nonkey Field	Definition
Routing Source AS	Key	Autonomous system of the source IP address.
Routing Destination AS	Key	Autonomous system of the destination IP address.
Routing Next-hop Address IPv6 BGP	Key	IPv6 address of the BGP next hop.
Interface Input	Key	Interface on which the traffic is received.
Interface Output	Key	Interface on which the traffic is transmitted.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Timestamp Sys-uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Timestamp Sys-uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

BGP Next-Hop ToS Predefined Record

The Flexible NetFlow "BGP next-hop ToS" predefined record creates flows based on BGP and ToS traffic flow data. The Flexible NetFlow "BGP next-hop ToS" predefined record uses the same key and nonkey fields as the original NetFlow "BGP next-hop ToS" aggregation cache.

Note This predefined record can be used to analyze only IPv4 traffic.

The table below lists the key and nonkey fields used in the "BGP next-hop ToS" predefined record.

Table 25: Key and Nonkey Fields Used by the Flexible NetFlow BGP Next-Hop ToS Predefined Record

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the ToS field.
IP Source autonomous system	Key	Autonomous system of the source IP address (peer or origin).
IP Destination autonomous system	Key	Autonomous system of the destination IP address (peer or origin).
IPv4 Next Hop Address BGP	Key	IPv4 address of the BGP next hop.

Field	Key or Nonkey Field	Definition
Interface Input	Key	Interface on which the traffic is received.
Interface Output	Key	Interface on which the traffic is transmitted.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Destination Prefix Predefined Record

The Flexible NetFlow "destination prefix" predefined record creates flows based on destination prefix traffic flow data. The Flexible NetFlow "destination prefix" predefined record uses the same key and nonkey fields as the original NetFlow "destination prefix" aggregation cache.

Note This predefined record can be used to analyze IPv4 and IPv6 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "destination prefix" predefined record.

Table 26: Key and Nonkey Fields Used by the Flexible NetFlow Destination Prefix Predefined Record

Field	Key or Nonkey Field	Definition
IP Destination autonomous system	Key	Autonomous system of the destination IP address (peer or origin).
IPv4 or IPv6 Destination Prefix	Key	Destination IP address ANDed with the destination prefix mask.
IPv4 or IPv6 Destination Mask	Key	Number of bits in the destination prefix.
Interface Output	Key	Interface on which the traffic is transmitted.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.

Field	Key or Nonkey Field	Definition
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Destination Prefix ToS Predefined Record

The Flexible NetFlow "destination prefix ToS" predefined record creates flows based on destination prefix and ToS traffic flow data. The Flexible NetFlow "destination prefix ToS" predefined record uses the same key and nonkey fields as the original NetFlow "destination prefix ToS" aggregation cache.

This predefined record is particularly useful for capturing data with which you can examine the destinations of network traffic passing through a NetFlow-enabled device.

Note This predefined record can be used to analyze only IPv4 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "destination prefix ToS" predefined record.

Table 27: Key and Nonkey Fields Used by the Flexible NetFlow Destination Prefix ToS Predefined Record

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the ToS field.
IP Destination autonomous system	Key	Autonomous system of the destination IP address (peer or origin).
IPv4 Destination Prefix	Key	Destination IP address ANDed with the destination prefix mask.
IPv4 Destination Mask	Key	Number of bits in the destination prefix.
Interface Output	Key	Interface on which the traffic is transmitted.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.

Field	Key or Nonkey Field	Definition
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Prefix Predefined Record

The Flexible NetFlow "prefix" predefined record creates flows based on the source and destination prefixes in the traffic flow data. The Flexible NetFlow "prefix" predefined record uses the same key and nonkey fields as the original NetFlow "prefix" aggregation cache.

Note This predefined record can be used to analyze IPv4 and IPv6 traffic. For IPv6 traffic, a minimum prefix mask length of 0 bits is assumed.

The table below lists the key and nonkey fields used in the Flexible NetFlow "prefix" predefined record.

Table 28: Key and Nonkey Fields Used by the Flexible NetFlow Prefix Predefined Record

Field	Key or Nonkey Field	Definition
IP Source autonomous system	Key	Autonomous system of the source IP address (peer or origin).
IP Destination autonomous system	Key	Autonomous system of the destination IP address (peer or origin).
IPv4 or IPv6 Source Prefix	Key	Source IP address ANDed with the source prefix mask, or the prefix to which the source IP address of the aggregated flows belongs.
IPv4 or IPv6 Source Mask	Key	Number of bits in the source prefix.
IPv4 or IPv6 Destination Prefix	Key	Destination IP address ANDed with the destination prefix mask.
IPv4 or IPv6 Destination Mask	Key	Number of bits in the destination prefix.
Interface Input	Key	Interface on which the traffic is received.
Interface Output	Key	Interface on which the traffic is transmitted.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.

Field	Key or Nonkey Field	Definition
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Prefix Port Predefined Record

The Flexible NetFlow "prefix port" predefined record creates flows based on source and destination prefixes and ports in the traffic flow data. The Flexible NetFlow "prefix port" predefined record uses the same key and nonkey fields as the original NetFlow "prefix port" aggregation cache.

This predefined record is particularly useful for capturing data with which you can examine the sources and destinations of network traffic passing through a NetFlow-enabled device.

Note This predefined record can be used to analyze only IPv4 traffic.

The table below lists the key and nonkey fields used in the destination Flexible NetFlow "prefix port" predefined record.

Table 29: Key and Nonkey Fields Used by the Flexible NetFlow Prefix Port Predefined Record

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the ToS field.
IP Protocol	Key	Value in the IP protocol field.
IPv4 Source Prefix	Key	Source IP address ANDed with the source prefix mask, or the prefix to which the source IP address of the aggregated flows belongs.
IPv4 Source Mask	Key	Number of bits in the source prefix.
IPv4 Destination Prefix	Key	Destination IP address ANDed with the destination prefix mask.
IPv4 Destination Mask	Key	Number of bits in the destination prefix.
Transport Source Port	Key	Value in the transport layer source port field.
Transport Destination Port	Key	Value in the transport layer destination port field.
Interface Input	Key	Interface on which the traffic is received.
Interface Output	Key	Interface on which the traffic is transmitted.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.

Field	Key or Nonkey Field	Definition
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Prefix ToS Predefined Record

The Flexible NetFlow "prefix ToS" predefined record creates flows based on source and destination prefixes and ToS traffic flow data. The Flexible NetFlow "prefix ToS" predefined record uses the same key and nonkey fields as the original NetFlow "destination prefix ToS" aggregation cache.

This predefined record is particularly useful for capturing data with which you can examine the sources and destinations of network traffic passing through a NetFlow-enabled device.

Note This predefined record can be used to analyze only IPv4 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "prefix ToS" predefined record.

Table 30: Key and Nonkey Fields Used by the Flexible NetFlow Prefix ToS Predefined Record

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the ToS field.
IP Source autonomous system	Key	Autonomous system of the source IP address (peer or origin).
IP Destination autonomous system	Key	Autonomous system of the destination IP address (peer or origin).
IPv4 Source Prefix	Key	Source IP address ANDed with the source prefix mask, or the prefix to which the source IP address of the aggregated flows belongs.
IPv4 Source Mask	Key	Number of bits in the source prefix.
IPv4 Destination Prefix	Key	Destination IP address ANDed with the destination prefix mask.
IPv4 Destination Mask	Key	Number of bits in the destination prefix.
Interface Input	Key	Interface on which the traffic is received.
Interface Output	Key	Interface on which the traffic is transmitted.

Field	Key or Nonkey Field	Definition
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Protocol Port Predefined Record

The Flexible NetFlow "protocol port" predefined record creates flows based on protocols and ports in the traffic flow data. The Flexible NetFlow "protocol port" predefined record uses the same key and nonkey fields as the original NetFlow "protocol port" aggregation cache.

Note This predefined record can be used to analyze IPv4 and IPv6 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "protocol port" predefined record.

Table 31: Key and Nonkey Fields Used by the Flexible NetFlow Protocol Port Predefined Record

Field	Key or Nonkey Field	Definition
IP Protocol	Key	Value in the IP protocol field.
Transport Source Port	Key	Value in the transport layer source port field.
Transport Destination Port	Key	Value in the transport layer destination port field.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Protocol Port ToS Predefined Record

The Flexible NetFlow "protocol port ToS" predefined record creates flows based on the protocol, port, and ToS value in the traffic data. The Flexible NetFlow "protocol port ToS" predefined record uses the same key and nonkey fields as the original NetFlow "protocol port ToS" aggregation cache.

This predefined record is particularly useful for capturing data with which you can examine network usage by type of traffic.

Note This predefined record can be used to analyze only IPv4 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "protocol port ToS" predefined record.

Table 32: Key and Nonkey Fields Used by the Flexible NetFlow Protocol Port ToS Predefined Record

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the ToS field.
IP Protocol	Key	Value in the IP protocol field.
Transport Source Port	Key	Value in the transport layer source port field.
Transport Destination Port	Key	Value in the transport layer destination port field.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Source Prefix Predefined Record

The Flexible NetFlow "source prefix" predefined record creates flows based on source prefixes in the network traffic. The Flexible NetFlow "source prefix" predefined record uses the same key and nonkey fields as the original NetFlow "source prefix" aggregation cache.

Note This predefined record can be used to analyze IPv4 and IPv6 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "source prefix" predefined record.

Table 33: Key and Nonkey Fields Used by the Flexible NetFlow Source Prefix Predefined Record

Field	Key or Nonkey Field	Definition
IP Source autonomous system	Key	Autonomous system of the source IP address (peer or origin).
IPv4 or IPv6 Source Prefix	Key	Source IP address ANDed with the source prefix mask, or the prefix to which the source IP address of the aggregated flows belongs.
IPv4 or IPv6 Source Mask	Key	Number of bits in the source prefix.
Interface Input	Key	Interface on which the traffic is received.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

Source Prefix ToS Predefined Record

The Flexible NetFlow "source prefix ToS" predefined record creates flows based on source prefixes and ToS values in the network traffic. The Flexible NetFlow "source prefix ToS" predefined record uses the same key and nonkey fields as the original NetFlow "source prefix ToS" aggregation cache.

This predefined record is particularly useful for capturing data with which you can examine the sources of network traffic passing through a NetFlow-enabled device.

Note This predefined record can be used to analyze only IPv4 traffic.

The table below lists the key and nonkey fields used in the Flexible NetFlow "source prefix ToS" predefined record.

Table 34: Key and Nonkey Fields Used by the Flexible NetFlow Source Prefix ToS Predefined Record

Field	Key or Nonkey Field	Definition
IP ToS	Key	Value in the ToS field.
IP Source autonomous system	Key	Autonomous system of the source IP address (peer or origin).

Field	Key or Nonkey Field	Definition
IPv4 Source Prefix	Key	Source IP address ANDed with the source prefix mask, or the prefix to which the source IP address of the aggregated flows belongs.
IPv4 Source Mask	Key	Number of bits in the source prefix.
Interface Input	Key	Interface on which the traffic is received.
Flow Direction	Key	Direction in which the flow is being monitored.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since this device was first booted) when the last packet was switched.

How to Configure Flexible Netflow

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ip** | **ipv6**} {**destination** | **source**} **address**
6. Repeat Step 5 as required to configure additional key fields for the record.
7. **match flow cts** {**source** | **destination**} **group-tag**
8.
9. Repeat the above step as required to configure additional nonkey fields for the record.
10. **end**
11. **show flow record** *record-name*

12. **show running-config flow record** *record-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record FLOW-RECORD-1`	Creates a flow record and enters Flexible NetFlow flow record configuration mode. • This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.
Step 5	**match** {**ip** | **ipv6**} {**destination** | **source**} **address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	**Note** This example configures the IPv4 destination address as a key field for the record. For information about the other key fields available for the **match ipv4** command, and the other **match** commands that are available to configure key fields.
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—

Step 7	**match flow cts** {**source** | **destination**} **group-tag**
	Example:
	`Device(config-flow-record)# match flow cts source group-tag`
	`Device(config-flow-record)# match flow cts destination group-tag`
		Note This example configures the CTS source group tag and destination group tag as a key field for the record. For information about the other key fields available for the **match ipv4**/**ipv6** command, and the other **match** commands that are available to configure key fields.

	Command or Action	Purpose
		Note • Ingress: • In an incoming packet, if a header is present, SGT will reflect the same value as the header. If no value is present, it will show zero. • The DGT value will not depend on the ingress port SGACL configuration. • Egress: • If either propagate SGT or CTS is disabled on the egress interface, then SGT will be zero. • In an outgoing packet, if SGACL configuration that corresponds to the (SGT, DGT) exists, DGT will be non-zero. • If SGACL is disabled on the egress port/VLAN or if global SGACL enforcement is disabled, then DGT will be zero
Step 8	**Example:**	Configures the input interface as a nonkey field for the record. **Note** This example configures the input interface as a nonkey field for the record.
Step 9	Repeat the above step as required to configure additional nonkey fields for the record.	—
Step 10	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.
Step 11	**show flow record** *record-name* **Example:** `Device# show flow record FLOW_RECORD-1`	(Optional) Displays the current status of the specified flow record.
Step 12	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Displaying the Current Status of a Flow Record

Perform this optional task to display the current status of a flow record.

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show flow record**

The **show flow record** command shows the current status of the flow monitor that you specify.

Example:

```
Device# show flow record

flow record FLOW-RECORD-2:
  Description:        Used for basic IPv6 traffic analysis
  No. of users:       1
  Total field space:  53 bytes
  Fields:
    match ipv6 destination address
    collect counter bytes
    collect counter packets
flow record FLOW-RECORD-1:
  Description:        Used for basic IPv4 traffic analysis
  No. of users:       1
  Total field space:  29 bytes
  Fields:
    match ipv4 destination address
    collect counter bytes
    collect counter packets
```

Verifying the Flow Record Configuration

Perform this optional task to verify the configuration commands that you entered.

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show running-config flow record**

The **show running-config flow record** command shows the configuration commands of the flow monitor that you specify.

Example:

```
Device# show running-config flow record

Current configuration:
!
flow record FLOW-RECORD-2
 description Used for basic IPv6 traffic analysis
 match ipv6 destination address
 collect counter bytes
 collect counter packets
!
flow record FLOW-RECORD-1
 description Used for basic IPv4 traffic analysis
 match ipv4 destination address
 collect counter bytes
 collect counter packets


!
```

Configuring a Flow Monitor for IPv4 or IPv6 Traffic Using the Predefined Record

To configure a flow monitor for IPv4/IPv6 traffic using the Flexible NetFlow "NetFlow IPv4/IPv6 original input" predefined record for the flow monitor, perform the following required task.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. The record format can be one of the predefined record formats, or an advanced user may create his or her own record format using the **collect** and **match**commands in Flexible NetFlow flow record configuration mode.

Note You must remove a flow monitor from all of the interfaces to which you have applied it before you can modify the **record** format of the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record netflow {ipv4 | ipv6} original-input**
6. **end**
7. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]]
8. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for monitoring IPv4 traffic`	(Optional) Creates a description for the flow monitor.
Step 5	**record netflow {ipv4 | ipv6} original-input** **Example:** `Device(config-flow-monitor)# record netflow ipv4 original-input`	Specifies the record for the flow monitor.
Step 6	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 7	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status and statistics for a Flexible NetFlow flow monitor.
Step 8	**show running-config flow monitor** *monitor-name* **Example:** `Device# show flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Configuring a Flow Exporter for the Flow Monitor

Perform this optional task to configure a flow exporter for the flow monitor in order to export the data that is collected by Flexible NetFlow to a remote system for further analysis and storage.

Flow exporters are used to send the data that you collect with Flexible NetFlow to a remote system such as a NetFlow Collection Engine. Exporters use UDP as the transport protocol and use the Version 9 export format.

Note Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.

You can export to a destination using either an IPv4 or IPv6 address.

Note When you configure an exporter, configure the exporter in such a way that the source interface is defined as a WAN interface. This configuration helps you prevent any unpredictable behavior because the NAT is not applied on the packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*]
6. **export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix**}
7. **transport udp** *udp-port*
8. **exit**
9. **flow monitor** *flow-monitor-name*
10. **exporter** *exporter-name*
11. **end**
12. **show flow exporter** *exporter-name*
13. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates a flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to datacenter`	(Optional) Creates a description for the flow exporter.
Step 5	**destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the hostname or IP address of the system to which the exporter sends data. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix**} **Example:** `Device(config-flow-exporter)# export-protocol netflow-v9`	Specifies the version of the NetFlow export protocol used by the exporter. • Default: **netflow-v9**.
Step 7	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 65`	Configures UDP as the transport protocol and specifies the UDP port on which the destination system is listening for exported Flexible NetFlow traffic.
Step 8	**exit** **Example:** `Device(config-flow-exporter)# exit`	Exits Flexible NetFlow flow exporter configuration mode and returns to global configuration mode.
Step 9	**flow monitor** *flow-monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Enters Flexible NetFlow flow monitor configuration mode for the flow monitor that you created previously.
Step 10	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	Specifies the name of an exporter that you created previously.
Step 11	**end** **Example:**	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-flow-monitor)# end`	
Step 12	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.
Step 13	**show running-config flow exporter** *exporter-name* **Example:** `Device<# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name*} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active**} *seconds* | { **normal** } **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 9	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:**	(Optional) Displays the status for a Flexible NetFlow flow monitor.

	Command or Action	Purpose
	`Device# show flow monitor FLOW-MONITOR-2 cache`	
Step 11	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Displaying the Current Status of a Flow Monitor

Perform this optional task to display the current status of a flow monitor.

SUMMARY STEPS

1. **enable**
2. **show flow monitor** *monitor-name*

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show flow monitor** *monitor-name*

The **show flow monitor** command shows the current status of the flow monitor that you specify.

Example:

```
Device# show flow monitor FLOW-MONITOR-1

Flow Monitor FLOW-MONITOR-1:
  Description:       Used for basic ipv4 traffic analysis
  Flow Record:       FLOW-RECORD-1
  Flow Exporter:     EXPORTER-1
  Cache:
    Type:              normal
    Status:            allocated

    Inactive Timeout:  15 secs
    Active Timeout:    1800 secs
    Update Timeout:    1800 secs
```

Displaying the Data in the Flow Monitor Cache

Perform this optional task to display the data in the flow monitor cache.

Before you begin

The interface on which you applied the input flow monitor must be receiving traffic that meets the criteria defined by the NetFlow original record before you can display the flows in the flow monitor cache.

SUMMARY STEPS

1. **enable**
2. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

Step 1 enable

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 show flow monitor name *monitor-name* cache format record

The **show flow monitor name** *monitor-name* **cache format record** command string displays the status, statistics, and flow data in the cache for a flow monitor.

Example:

```
Device# show flow monitor name FLOW-MONITOR-1 cache format record

Cache type:                               Normal

  Current entries:                           4
  High Watermark:                            4
  Flows added:                             101
  Flows aged:                               97
    - Active timeout    (  1800 secs)        3
    - Inactive timeout (    15 secs)        94
    - Event aged                             0
    - Watermark aged                         0
    - Emergency aged                         0
IPV4 DESTINATION ADDRESS:  172.16.10.5
ipv4 source address:       10.10.11.1
trns source port:          25
trns destination port:     25
counter bytes:             72840
counter packets:           1821
IPV4 DESTINATION ADDRESS:  172.16.10.2
ipv4 source address:       10.10.10.2
trns source port:          20
trns destination port:     20
counter bytes:             3913860
counter packets:           7326
IPV4 DESTINATION ADDRESS:  172.16.10.200
```

```
ipv4 source address:       192.168.67.6
trns source port:          0
trns destination port:     3073
counter bytes:             51072
counter packets:           1824

Device# show flow monitor name FLOW-MONITOR-2 cache format record

Cache type:                              Normal

  Current entries:                         2
  High Watermark:                          3
  Flows added:                            95
  Flows aged:                             93
    - Active timeout   (  1800 secs)       0
    - Inactive timeout (    15 secs)      93
    - Event aged                           0
    - Watermark aged                       0
    - Emergency aged                       0
IPV6 DESTINATION ADDRESS:  2001:DB8:4:ABCD::2
ipv6 source address:       2001:DB8:1:ABCD::1
trns source port:          33572
trns destination port:     23
counter bytes:             19140
counter packets:           349
IPV6 DESTINATION ADDRESS:  FF02::9
ipv6 source address:       FE80::A8AA:BBFF:FEBB:CC03
trns source port:          521
trns destination port:     521
counter bytes:             92
counter packets:           1
```

Verifying the Flow Monitor Configuration

Perform this optional task to verify the configuration commands that you entered.

SUMMARY STEPS

1. **enable**
2. **show running-config flow monitor**

DETAILED STEPS

Step 1 enable

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 show running-config flow monitor

The **show running-config flow monitor** command shows the configuration commands of the flow monitor that you specify.

Example:

```
Device# show running-config flow monitor FLOW-MONITOR-1

Current configuration:
!
flow monitor FLOW-MONITOR-1
 description Used for basic ipv4 traffic analysis
 record FLOW-RECORD-1
 exporter EXPORTER-1

!
```

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

While running the **ip flow monitor** command for the first interface to enable FNF monitor, you may see the following warning message displaying a GLOBAL memory allocation failure. This log is triggered by enabling FNF monitoring with a large cache size.

```
Jul  4 01:45:00.255: %CPPEXMEM-3-NOMEM: F0/0: cpp_cp_svr: QFP: 0, GLOBAL memory allocation
 of 90120448 bytes by FNF failed
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: CPR STILE
EXMEM GRAPH, Allocations: 877, Type: GLOBAL
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: SBC, Bytes
Allocated: 53850112, Type: GLOBAL
```

The warning message does not necessarily indicate a flow monitor application failure. The warning message can indicate internal steps that FNF uses for applying memory from the EXMEM infrastructure.

To ensure that the FNF monitor is enabled successfully, use the **show flow monitor** *monitor-name* command to check **Status** (**allocated** or **not allocated**) of a flow monitor. For more information, see Displaying the Current Status of a Flow Monitor, on page 156.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Verifying That Flexible NetFlow Is Enabled on an Interface

Perform this optional task to verify that Flexible NetFlow is enabled on an interface.

SUMMARY STEPS

1. **enable**

2. **show flow interface** *type number*

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show flow interface** *type number*

The **show flow interface** command verifies that Flexible NetFlow is enabled on an interface.

Example:

```
Device# show flow interface GigabitEthernet 0/0/0

Interface GigabitEthernet0/0/0
  FNF:  monitor:          FLOW-MONITOR-1
        direction:        Input
        traffic(ip):      on
  FNF:  monitor:          FLOW-MONITOR-2
        direction:        Input
        traffic(ipv6):    on
Device# show flow interface GigabitEthernet 1/0/0
Interface GigabitEthernet1/0/0
  FNF:  monitor:          FLOW-MONITOR-1
        direction:        Output
        traffic(ip):      on
  FNF:  monitor:          FLOW-MONITOR-2
        direction:        Input
        traffic(ipv6):    on
```

Configuration Examples for Flexible Netflow

Example: Configuring a Flexible NetFlow Predefined Record for IPv4 Traffic

The following example shows how to configure a flow monitor using the Flexible NetFlow "BGP ToS next-hop" predefined record to monitor IPv4 traffic.

This sample starts in global configuration mode:

```
!
flow monitor FLOW-MONITOR-1
 record netflow ipv4 bgp-nexthop-tos
 exit
!
ip cef
!
```

```
interface Ethernet 0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

Example: Configuring a Flexible NetFlow Predefined Record for IPv6 Traffic

The following example shows how to configure a flow monitor using the Flexible NetFlow "source prefix" predefined record to monitor IPv6 traffic.

This sample starts in global configuration mode:

```
!
flow monitor FLOW-MONITOR-2
 record netflow ipv6 source-prefix
 exit
ip cef
ipv6 cef
!
interface GigabitEthernet 0/0/0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ipv6 flow monitor FLOW-MONITOR-2 input
!
```

Example: Configuring a Normal Flow Record Cache with a Limited Number of Flows

The following example is designed to monitor the type of service (ToS) field usage on all interfaces in the router. An exporter is not configured because this example is intended to be used to capture additional data for analysis on the router using the **show flow monitor** command.

This example starts in global configuration mode.

```
!
flow record QOS_RECORD
 description UD: Flow Record to monitor the use of TOS within this router/network
 match interface input
 match interface output
 match ipv4 tos
 collect counter packets
 collect counter bytes
 exit
!
flow monitor QOS_MONITOR
 description UD: Flow Monitor which watches the limited combinations of interface and TOS
 record QOS_RECORD
 cache type normal
 cache entries 8192   ! 2^5 (combos of interfaces) * 256 (values of TOS)
 exit
!
interface GigabitEthernet0/0/0
 ip flow monitor QOS_MONITOR input
 exit
!
interface GigabitEthernet0/1/0
 ip flow monitor QOS_MONITOR input
 exit
!
```

```
interface GigabitEthernet0/2/0
 ip flow monitor QOS_MONITOR input
 exit
!
```

The display from the **show flow monitor** command shows the current status of the cache.

```
Router# show flow monitor QOS_MONITOR cache

  Cache type:            Normal
  Cache size:                          8192
  Current entries:                        2
  High Watermark:                         2
  Flows added:                            2
  Updates sent         (  1800 secs)      0
```

Example: Configuring a Customized Flow Record Cache for Monitoring IPv6 Traffic

The following example creates a customized flow record cache for monitoring IPv6 traffic.

This example starts in global configuration mode.

Example: Configuring a Permanent Flow Record Cache with a Limited Number of Flows

The following example is designed to monitor the type of service (ToS) field usage on all interfaces in the router. An exporter is not configured because this example is intended to be used to capture additional data for analysis on the router using the **show flow monitor**command.

This example starts in global configuration mode.

```
!
ip cef
!
flow record QOS_RECORD
 description UD: Flow Record to monitor the use of TOS within this router/network
 match interface input
 match interface output
 match ipv4 tos
 collect counter packets
 collect counter bytes
 exit
!
flow monitor QOS_MONITOR
 description UD: Flow Monitor which watches the limited combinations of interface and TOS
 record QOS_RECORD
 cache type permanent
 cache entries 8192   ! 2^5 (combos of interfaces) * 256 (values of TOS)
 exit
!
interface ethernet0/0
 ip flow monitor QOS_MONITOR input
 exit
!
interface ethernet0/1
 ip flow monitor QOS_MONITOR input
```

```
 exit
!
interface ethernet0/2
 ip flow monitor QOS_MONITOR input
 exit
!
interface serial2/0
 ip flow monitor QOS_MONITOR input
 exit
!
interface serial2/1
 ip flow monitor QOS_MONITOR input
!
```

The display from the **show flow monitor** command shows the current status of the cache.

```
Router# show flow monitor QOS_MONITOR cache
  Cache type:                           Permanent
  Cache size:                                8192
  Current entries:                              2
  High Watermark:                               2
  Flows added:                                  2
  Updates sent         (  1800 secs)            0
```

Example: Configuring Flexible NetFlow Egress Accounting for IPv4 and IPv6 Traffic

The following example shows how to configure Flexible NetFlow egress accounting for IPv4 and IPv6 traffic.

This example starts in global configuration mode.

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exit
```

```
!
ip cef
ipv6 cef
!
interface GigabitEthernet0/0/0
 ip address 172.16.6.2 255.255.255.0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ip flow monitor FLOW-MONITOR-1 output
 ipv6 flow monitor FLOW-MONITOR-2 output
!
```

Example: Configuring Flexible NetFlow Subinterface Support

The following example shows how to configure Flexible NetFlow subinterface support for IPv4 traffic.

This example starts in global configuration mode.

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
ip cef
!
interface Ethernet0/0.1
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

The following example shows how to configure Flexible NetFlow to emulate NetFlow subinterface support for IPv6 traffic.

This sample starts in global configuration mode:

```
!
flow record v6_r1

match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long

!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exit
!
ip cef
```

```
ipv6 cef
!
interface Ethernet0/0.1
 ipv6 address 2001:DB8:2:ABCD::2/48
 ipv6 flow monitor FLOW-MONITOR-2 input
!
```

Example: Configuring Flexible NetFlow Multiple Export Destinations

The following example shows how to configure Flexible NetFlow multiple export destinations.

This example starts in global configuration mode.

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 transport udp 90
 exit
!
flow exporter EXPORTER-2
 destination 172.16.10.3
 transport udp 90
 exit
!
flow monitor FLOW-MONITOR-1
 record netflow-original
 exporter EXPORTER-2
 exporter EXPORTER-1
 exit
!
ip cef
!
interface GigabitEthernet0/0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

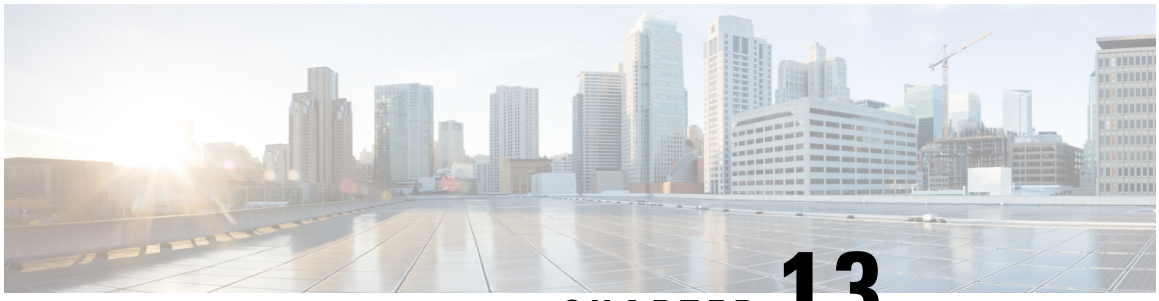
Feature Information for Flexible NetFlow

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 35: Feature Information for Flexible NetFlow

Feature Name	Releases	Feature Information
Flexible NetFlow	12.2(33)SRC 12.2(50)SY 12.4(9)T 15.0(1)SY 15.0(1)SY1 Cisco IOS XE Release 3.1S	Flexible NetFlow is introduced. Support for this feature was added for Cisco 7200 series routers in Cisco IOS Release 12.2(33)SRC. The following commands were introduced or modified: **cache** (Flexible NetFlow), **clear flow exporter**, **clear flow monitor**, **clear sampler**, **collect counter**, **collect flow**, **collect interface**, **collect ipv4**, **collect ipv4 destination**, **collect ipv4 fragmentation**, **collect ipv4 section**, **collect ipv4 source**, **collect ipv4 total-length**, **collect ipv4 ttl**, **collect routing**, **collect timestamp sys-uptime**, **collect transport**, **collect transport icmp ipv4**, **collect transport tcp**, **collect transport udp**, **debug flow exporter**, **debug flow monitor**, **debug flow record**, **debug sampler**, **description** (Flexible NetFlow), **destination**, **dscp** (Flexible NetFlow), **exporter**, **flow exporter**, **flow monitor**, **flow platform**, **flow record**, **ip flow monitor**, **match flow**, **match interface** (Flexible NetFlow), **match ipv4**, **match ipv4 destination**, **match ipv4 fragmentation**, **match ipv4 section**, **match ipv4 source**, **match ipv4 total-length**, **match ipv4 ttl**, **match routing**, **match transport**, **match transport icmp ipv4**, **match transport tcp**, **match transport udp**, **mode** (Flexible NetFlow), **option** (Flexible NetFlow), **record**, **sampler**, **show flow exporter**, **show flow interface**, **show flow monitor**, **show flow record**, **show sampler**, **source** (Flexible NetFlow), **statistics packet**, **template data timeout**, **transport** (Flexible NetFlow).



CHAPTER 13

Flexible NetFlow—IPv4 Unicast Flows

The Flexible Netflow—IPv4 Unicast Flows feature enables Flexible NetFlow to monitor IPv4 traffic.

Information About Flexible NetFlow IPv4 Unicast Flows

Flexible NetFlow—IPv4 Unicast Flows Overview

This feature enables Flexible NetFlow to monitor IPv4 traffic.

How to Configure Flexible NetFlow IPv4 Unicast Flows

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ip** | **ipv6**} {**destination** | **source**} **address**

6. Repeat Step 5 as required to configure additional key fields for the record.
7. **match flow cts** {**source** | **destination**} **group-tag**
8.
9. Repeat the above step as required to configure additional nonkey fields for the record.
10. **end**
11. **show flow record** *record-name*
12. **show running-config flow record** *record-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record FLOW-RECORD-1`	Creates a flow record and enters Flexible NetFlow flow record configuration mode. • This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.
Step 5	**match** {**ip** | **ipv6**} {**destination** | **source**} **address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	**Note** This example configures the IPv4 destination address as a key field for the record. For information about the other key fields available for the **match ipv4** command, and the other **match** commands that are available to configure key fields.
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—
Step 7	**match flow cts** {**source** | **destination**} **group-tag** **Example:** `Device(config-flow-record)# match flow cts source group-tag`	**Note** This example configures the CTS source group tag and destination group tag as a key field for the record. For information about the other key fields available for the **match ipv4** / **ipv6** command, and the other **match** commands that are available to configure key fields.

	Command or Action	Purpose
	`Device(config-flow-record)# match flow cts destination group-tag`	**Note** • Ingress: • In an incoming packet, if a header is present, SGT will reflect the same value as the header. If no value is present, it will show zero. • The DGT value will not depend on the ingress port SGACL configuration. • Egress: • If either propagate SGT or CTS is disabled on the egress interface, then SGT will be zero. • In an outgoing packet, if SGACL configuration that corresponds to the (SGT, DGT) exists, DGT will be non-zero. • If SGACL is disabled on the egress port/VLAN or if global SGACL enforcement is disabled, then DGT will be zero
Step 8	**Example:**	Configures the input interface as a nonkey field for the record. **Note** This example configures the input interface as a nonkey field for the record.
Step 9	Repeat the above step as required to configure additional nonkey fields for the record.	—
Step 10	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.
Step 11	**show flow record** *record-name* **Example:** `Device# show flow record FLOW_RECORD-1`	(Optional) Displays the current status of the specified flow record.
Step 12	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Configuring the Flow Exporter

Perform this required task to configure the flow exporter.

Note Each flow exporter supports only one destination.

You can export to a destination using either an IPv4 or IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
6. **dscp** *dscp*
7. **source** *interface-type interface-number*
8. **output-features**
9. **template data timeout** *seconds*
10. **transport udp** *udp-port*
11. **ttl** *seconds*
12. **end**
13. **show flow exporter** *exporter-name*
14. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates the flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.

	Command or Action	Purpose
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to the datacenter`	(Optional) Configures a description to the exporter that will appear in the configuration and the display of the **show flow exporter** command.
Step 5	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the destination system for the exporter. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**dscp** *dscp* **Example:** `Device(config-flow-exporter)# dscp 63`	(Optional) Configures differentiated services code point (DSCP) parameters for datagrams sent by the exporter. • The range for the *dscp* argument is from 0 to 63. Default: 0.
Step 7	**source** *interface-type interface-number* **Example:** `Device(config-flow-exporter)# source ethernet 0/0`	(Optional) Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 8	**output-features** **Example:** `Device(config-flow-exporter)# output-features`	(Optional) Enables sending export packets using quality of service (QoS) and encryption.
Step 9	**template data timeout** *seconds* **Example:** `Device(config-flow-exporter)# template data timeout 120`	(Optional) Configures resending of templates based on a timeout. • The range for the *seconds* argument is 1 to 86400 (86400 seconds = 24 hours).
Step 10	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 650`	Specifies the UDP port on which the destination system is listening for exported datagrams. • The range for the *udp-port* argument is from 1 to 65536.
Step 11	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 15`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. • The range for the *seconds* argument is from 1 to 255.
Step 12	**end** **Example:** `Device(config-flow-exporter)# end`	Exits flow exporter configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 13	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.
Step 14	**show running-config flow exporter** *exporter-name* **Example:** `Device# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name*} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active**} *seconds* | { **normal** } **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 9	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:**	(Optional) Displays the status for a Flexible NetFlow flow monitor.

	Command or Action	Purpose
	`Device# show flow monitor FLOW-MONITOR-2 cache`	
Step 11	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

While running the **ip flow monitor** command for the first interface to enable FNF monitor, you may see the following warning message displaying a GLOBAL memory allocation failure. This log is triggered by enabling FNF monitoring with a large cache size.

```
Jul  4 01:45:00.255: %CPPEXMEM-3-NOMEM: F0/0: cpp_cp_svr: QFP: 0, GLOBAL memory allocation
 of 90120448 bytes by FNF failed
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: CPR STILE
EXMEM GRAPH, Allocations: 877, Type: GLOBAL
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: SBC, Bytes
Allocated: 53850112, Type: GLOBAL
```

The warning message does not necessarily indicate a flow monitor application failure. The warning message can indicate internal steps that FNF uses for applying memory from the EXMEM infrastructure.

To ensure that the FNF monitor is enabled successfully, use the **show flow monitor** *monitor-name* command to check **Status** (**allocated** or **not allocated**) of a flow monitor. For more information, see Displaying the Current Status of a Flow Monitor, on page 156.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Configuring and Enabling Flexible NetFlow with Data Export

You must create a flow monitor to configure the types of traffic for which you want to export the cache data. You must enable the flow monitor by applying it to at least one interface to start exporting data. To configure and enable Flexible NetFlow with data export, perform this required task.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. The record format can be one of the predefined record formats, or an

advanced user may create his or her own record format using the **collect** and **match**commands in Flexible NetFlow flow record configuration mode.

Note You must remove a flow monitor from all of the interfaces to which you have applied it before you can modify the **record** format of the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6** *record* [**peer**] }]
5. **exporter** *exporter-name*
6. **exit**
7. **interface** *type number*
8. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6** *record* [**peer**] }] **Example:** `Device(config-flow-monitor)# record netflow ipv4 original-input`	Specifies the record for the flow monitor.

	Command or Action	Purpose
Step 5	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	Specifies the name of an exporter that you created previously.
Step 6	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 8	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates the flow monitor that you created previously by assigning it to the interface to analyze traffic.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status and statistics for a Flexible NetFlow flow monitor. This will verify data export is enabled for the flow monitor cache.

Configuration Examples for Flexible NetFlow IPv4 Unicast Flows

Example: Configuring Multiple Export Destinations

The following example shows how to configure multiple export destinations for Flexible NetFlow for IPv4 or IPv6 traffic.

This sample starts in global configuration mode:

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 transport udp 90
```

```
 exit
!
flow exporter EXPORTER-2
 destination 172.16.10.3
 transport udp 90
 exit
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!

flow monitor FLOW-MONITOR-1
 record v4_r1
 exporter EXPORTER-2
 exporter EXPORTER-1
!
!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exporter EXPORTER-2
 exporter EXPORTER-1
!
ip cef
!
interface GigabitEthernet1/0/0
 ip address 172.16.6.2 255.255.255.0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ip flow monitor FLOW-MONITOR-1 input
 ipv6 flow monitor FLOW-MONITOR-2 input
!
```

The following display output shows that the flow monitor is exporting data to the two exporters:

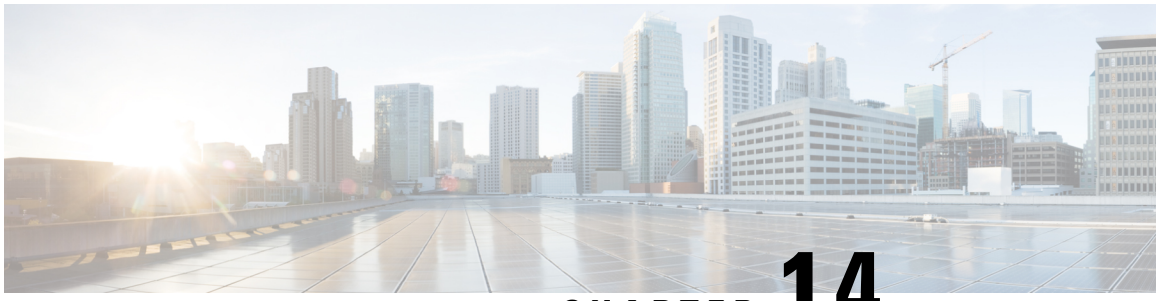
```
Device# show flow monitor FLOW-MONITOR-1
Flow Monitor FLOW-MONITOR-1:
  Description:       User defined
  Flow Record:       v4_r1
  Flow Exporter:     EXPORTER-1
                     EXPORTER-2
  Cache:
    Type:              normal (Platform cache)
    Status:            allocated
    Size:              4096 entries / 311316 bytes
    Inactive Timeout:  15 secs
    Active Timeout:    1800 secs
    Update Timeout:    1800 secs
```

Example: Configuring Flexible NetFlow Egress Accounting for IPv4 and IPv6 Traffic

The following example shows how to configure Flexible NetFlow egress accounting for IPv4 and IPv6 traffic.

This example starts in global configuration mode.

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exit
!
ip cef
ipv6 cef
!
interface GigabitEthernet0/0/0
 ip address 172.16.6.2 255.255.255.0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ip flow monitor FLOW-MONITOR-1 output
 ipv6 flow monitor FLOW-MONITOR-2 output
!
```



CHAPTER 14

Flexible NetFlow—IPv6 Unicast Flows

The Flexible NetFlow—IPv6 Unicast Flows feature enables Flexible NetFlow to monitor IPv6 traffic.

Information About Flexible NetFlow IPv6 Unicast Flows

Flexible NetFlow IPv6 Unicast Flows Overview

This feature enables Flexible NetFlow to monitor IPv6 traffic.

How to Configure Flexible NetFlow IPv6 Unicast Flows

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ip** | **ipv6**} {**destination** | **source**} **address**

6. Repeat Step 5 as required to configure additional key fields for the record.
7. **match flow cts** {**source** | **destination**} **group-tag**
8.
9. Repeat the above step as required to configure additional nonkey fields for the record.
10. **end**
11. **show flow record** *record-name*
12. **show running-config flow record** *record-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record FLOW-RECORD-1`	Creates a flow record and enters Flexible NetFlow flow record configuration mode. • This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.
Step 5	**match** {**ip** | **ipv6**} {**destination** | **source**} **address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	**Note** This example configures the IPv4 destination address as a key field for the record. For information about the other key fields available for the **match ipv4** command, and the other **match** commands that are available to configure key fields.
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—
Step 7	**match flow cts** {**source** | **destination**} **group-tag** **Example:** `Device(config-flow-record)# match flow cts source group-tag`	**Note** This example configures the CTS source group tag and destination group tag as a key field for the record. For information about the other key fields available for the **match ipv4** / **ipv6** command, and the other **match** commands that are available to configure key fields.

	Command or Action	Purpose
	`Device(config-flow-record)# match flow cts destination group-tag`	**Note** • Ingress: • In an incoming packet, if a header is present, SGT will reflect the same value as the header. If no value is present, it will show zero. • The DGT value will not depend on the ingress port SGACL configuration. • Egress: • If either propagate SGT or CTS is disabled on the egress interface, then SGT will be zero. • In an outgoing packet, if SGACL configuration that corresponds to the (SGT, DGT) exists, DGT will be non-zero. • If SGACL is disabled on the egress port/VLAN or if global SGACL enforcement is disabled, then DGT will be zero
Step 8	**Example:**	Configures the input interface as a nonkey field for the record. **Note** This example configures the input interface as a nonkey field for the record.
Step 9	Repeat the above step as required to configure additional nonkey fields for the record.	—
Step 10	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.
Step 11	**show flow record** *record-name* **Example:** `Device# show flow record FLOW_RECORD-1`	(Optional) Displays the current status of the specified flow record.
Step 12	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Configuring the Flow Exporter

Perform this required task to configure the flow exporter.

Note Each flow exporter supports only one destination.

You can export to a destination using either an IPv4 or IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
6. **dscp** *dscp*
7. **source** *interface-type interface-number*
8. **output-features**
9. **template data timeout** *seconds*
10. **transport udp** *udp-port*
11. **ttl** *seconds*
12. **end**
13. **show flow exporter** *exporter-name*
14. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates the flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.

	Command or Action	Purpose
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to the datacenter`	(Optional) Configures a description to the exporter that will appear in the configuration and the display of the **show flow exporter** command.
Step 5	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the destination system for the exporter. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**dscp** *dscp* **Example:** `Device(config-flow-exporter)# dscp 63`	(Optional) Configures differentiated services code point (DSCP) parameters for datagrams sent by the exporter. • The range for the *dscp* argument is from 0 to 63. Default: 0.
Step 7	**source** *interface-type interface-number* **Example:** `Device(config-flow-exporter)# source ethernet 0/0`	(Optional) Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 8	**output-features** **Example:** `Device(config-flow-exporter)# output-features`	(Optional) Enables sending export packets using quality of service (QoS) and encryption.
Step 9	**template data timeout** *seconds* **Example:** `Device(config-flow-exporter)# template data timeout 120`	(Optional) Configures resending of templates based on a timeout. • The range for the *seconds* argument is 1 to 86400 (86400 seconds = 24 hours).
Step 10	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 650`	Specifies the UDP port on which the destination system is listening for exported datagrams. • The range for the *udp-port* argument is from 1 to 65536.
Step 11	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 15`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. • The range for the *seconds* argument is from 1 to 255.
Step 12	**end** **Example:** `Device(config-flow-exporter)# end`	Exits flow exporter configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 13	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.
Step 14	**show running-config flow exporter** *exporter-name* **Example:** `Device# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name*} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active**} *seconds* | { **normal** } **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 9	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:**	(Optional) Displays the status for a Flexible NetFlow flow monitor.

	Command or Action	Purpose
	`Device# show flow monitor FLOW-MONITOR-2 cache`	
Step 11	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

While running the **ip flow monitor** command for the first interface to enable FNF monitor, you may see the following warning message displaying a GLOBAL memory allocation failure. This log is triggered by enabling FNF monitoring with a large cache size.

```
Jul  4 01:45:00.255: %CPPEXMEM-3-NOMEM: F0/0: cpp_cp_svr: QFP: 0, GLOBAL memory allocation
 of 90120448 bytes by FNF failed
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: CPR STILE
EXMEM GRAPH, Allocations: 877, Type: GLOBAL
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: SBC, Bytes
Allocated: 53850112, Type: GLOBAL
```

The warning message does not necessarily indicate a flow monitor application failure. The warning message can indicate internal steps that FNF uses for applying memory from the EXMEM infrastructure.

To ensure that the FNF monitor is enabled successfully, use the **show flow monitor** *monitor-name* command to check **Status** (**allocated** or **not allocated**) of a flow monitor. For more information, see Displaying the Current Status of a Flow Monitor, on page 156.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Configuring and Enabling Flexible NetFlow with Data Export

You must create a flow monitor to configure the types of traffic for which you want to export the cache data. You must enable the flow monitor by applying it to at least one interface to start exporting data. To configure and enable Flexible NetFlow with data export, perform this required task.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. The record format can be one of the predefined record formats, or an

advanced user may create his or her own record format using the **collect** and **match**commands in Flexible NetFlow flow record configuration mode.

Note You must remove a flow monitor from all of the interfaces to which you have applied it before you can modify the **record** format of the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6** *record* [**peer**] }]
5. **exporter** *exporter-name*
6. **exit**
7. **interface** *type number*
8. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6** *record* [**peer**] }] **Example:** `Device(config-flow-monitor)# record netflow ipv4 original-input`	Specifies the record for the flow monitor.

	Command or Action	Purpose
Step 5	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	Specifies the name of an exporter that you created previously.
Step 6	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 8	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates the flow monitor that you created previously by assigning it to the interface to analyze traffic.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status and statistics for a Flexible NetFlow flow monitor. This will verify data export is enabled for the flow monitor cache.

Configuration Examples for Flexible NetFlow IPv6 Unicast Flows

Example: Configuring Multiple Export Destinations

The following example shows how to configure multiple export destinations for Flexible NetFlow for IPv4 or IPv6 traffic.

This sample starts in global configuration mode:

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 transport udp 90
```

```
 exit
!
flow exporter EXPORTER-2
 destination 172.16.10.3
 transport udp 90
 exit
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!

flow monitor FLOW-MONITOR-1
 record v4_r1
 exporter EXPORTER-2
 exporter EXPORTER-1
!
!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exporter EXPORTER-2
 exporter EXPORTER-1
!
ip cef
!
interface GigabitEthernet1/0/0
 ip address 172.16.6.2 255.255.255.0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ip flow monitor FLOW-MONITOR-1 input
 ipv6 flow monitor FLOW-MONITOR-2 input
!
```

The following display output shows that the flow monitor is exporting data to the two exporters:

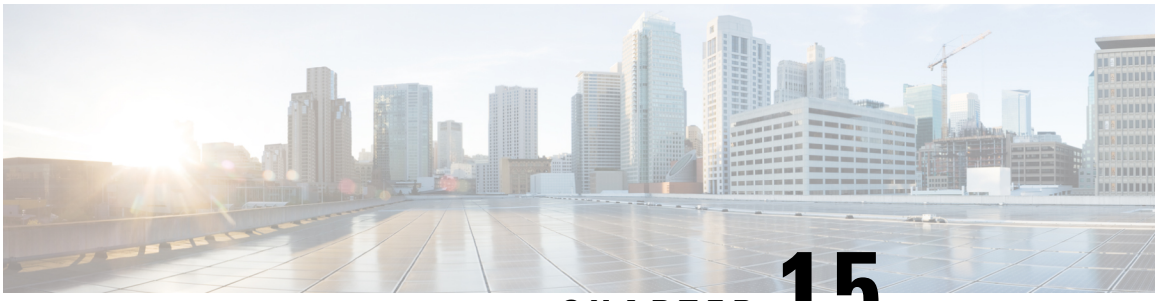
```
Device# show flow monitor FLOW-MONITOR-1
Flow Monitor FLOW-MONITOR-1:
  Description:       User defined
  Flow Record:       v4_r1
  Flow Exporter:     EXPORTER-1
                     EXPORTER-2
  Cache:
    Type:              normal (Platform cache)
    Status:            allocated
    Size:              4096 entries / 311316 bytes
    Inactive Timeout:  15 secs
    Active Timeout:    1800 secs
    Update Timeout:    1800 secs
```

Example: Configuring Flexible NetFlow Egress Accounting for IPv4 and IPv6 Traffic

The following example shows how to configure Flexible NetFlow egress accounting for IPv4 and IPv6 traffic.

This example starts in global configuration mode.

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exit
!
ip cef
ipv6 cef
!
interface GigabitEthernet0/0/0
 ip address 172.16.6.2 255.255.255.0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ip flow monitor FLOW-MONITOR-1 output
 ipv6 flow monitor FLOW-MONITOR-2 output
!
```



CHAPTER 15

Flexible NetFlow—MPLS Egress NetFlow

The Flexible NetFlow--MPLS Egress NetFlow feature allows you to capture IP flow information for packets undergoing MPLS label disposition; that is, packets that arrive on a router as MPLS packets and are transmitted as IP packets.

Information About Flexible NetFlow MPLS Egress NetFlow

Flexible NetFlow MPLS Egress NetFlow

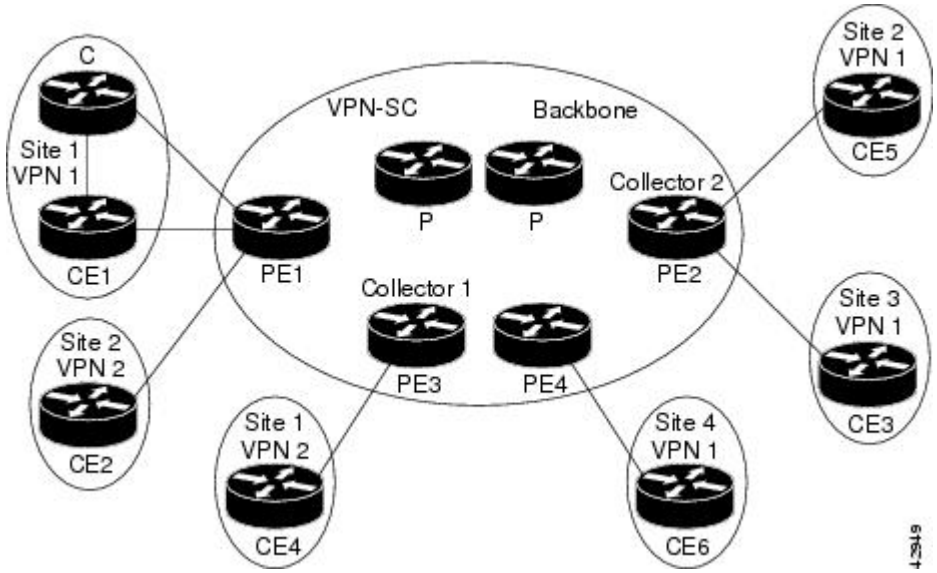
The Flexible NetFlow - MPLS Egress NetFlow feature allows you to capture IP flow information for packets that arrive on a router as Multiprotocol Label Switching (MPLS) packets and are transmitted as IP packets. This feature allows you to capture the MPLS VPN IP flows that are traveling through the service provider backbone from one site of a VPN to another site of the same VPN. The Flexible NetFlow - MPLS Egress NetFlow feature is enabled by applying a flow monitor in output (egress) mode on the provider edge (PE) to customer edge (CE) interface of the provider's network.

The figure below shows a sample MPLS VPN network topology that includes four VPN 1 sites and two VPN 2 sites. If the Flexible NetFlow - MPLS Egress NetFlow is enabled on an outgoing PE interface by applying a flow monitor in output mode, IP flow information for packets that arrive at the PE as MPLS packets (from an MPLS VPN) and that are transmitted as IP packets to the PE router is captured. For example:

- To capture the flow of traffic going to site 2 of VPN 1 from any remote VPN 1 sites, you enable a flow monitor in output mode on link PE2-CE5 of provider edge router PE2.
- To capture the flow of traffic going to site 1 of VPN 2 from any remote VPN 2 site, you enable a flow monitor in output mode on link PE3-CE4 of the provider edge router PE3.

The flow data is stored in the Flexible NetFlow cache. You can use the **show flow monitor** *monitor-name* **cache**command to display the flow data in the cache.

Figure 8: Sample MPLS VPN Network Topology with Flexible NetFlow--MPLS Egress NetFlow Feature



If you configure a Flexible NetFlow exporter for the flow monitors you use for the Flexible NetFlow - MPLS Egress NetFlow feature, the PE routers will export the captured flows to the configured collector devices in the provider network. Applications such as the Network Data Analyzer or the VPN Solution Center (VPN-SC) can gather information from the captured flows and compute and display site-to-site VPN traffic statistics.

Limitations

When using Flexible NetFlow to monitor outbound traffic on a router at the edge of an MPLS cloud, for IP traffic that leaves over a VRF, the following fields are not collected and have a value of 0:

- destination mask
- destination prefix
- destination AS numbers
- destination BGP traffic index
- nexthop
- BGP nexthop

How to Configure Flexible NetFlow MPLS Egress NetFlow

Configuring a Flow Exporter for the Flow Monitor

Perform this optional task to configure a flow exporter for the flow monitor in order to export the data that is collected by Flexible NetFlow to a remote system for further analysis and storage.

Flow exporters are used to send the data that you collect with Flexible NetFlow to a remote system such as a NetFlow Collection Engine. Exporters use UDP as the transport protocol and use the Version 9 export format.

Note Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.

You can export to a destination using either an IPv4 or IPv6 address.

Note When you configure an exporter, configure the exporter in such a way that the source interface is defined as a WAN interface. This configuration helps you prevent any unpredictable behavior because the NAT is not applied on the packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*]
6. **export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix**}
7. **transport udp** *udp-port*
8. **exit**
9. **flow monitor** *flow-monitor-name*
10. **exporter** *exporter-name*
11. **end**
12. **show flow exporter** *exporter-name*
13. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:**	Creates a flow exporter and enters Flexible NetFlow flow exporter configuration mode.

	Command or Action	Purpose
	`Device(config)# flow exporter EXPORTER-1`	• This command also allows you to modify an existing flow exporter.
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to datacenter`	(Optional) Creates a description for the flow exporter.
Step 5	**destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the hostname or IP address of the system to which the exporter sends data. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix**} **Example:** `Device(config-flow-exporter)# export-protocol netflow-v9`	Specifies the version of the NetFlow export protocol used by the exporter. • Default: **netflow-v9**.
Step 7	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 65`	Configures UDP as the transport protocol and specifies the UDP port on which the destination system is listening for exported Flexible NetFlow traffic.
Step 8	**exit** **Example:** `Device(config-flow-exporter)# exit`	Exits Flexible NetFlow flow exporter configuration mode and returns to global configuration mode.
Step 9	**flow monitor** *flow-monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Enters Flexible NetFlow flow monitor configuration mode for the flow monitor that you created previously.
Step 10	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	Specifies the name of an exporter that you created previously.
Step 11	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 12	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.
Step 13	**show running-config flow exporter** *exporter-name* **Example:** `Device<# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name*} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active**} *seconds* | { **normal** } **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 9	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:**	(Optional) Displays the status for a Flexible NetFlow flow monitor.

	Command or Action	Purpose
	`Device# show flow monitor FLOW-MONITOR-2 cache`	
Step 11	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

While running the **ip flow monitor** command for the first interface to enable FNF monitor, you may see the following warning message displaying a GLOBAL memory allocation failure. This log is triggered by enabling FNF monitoring with a large cache size.

```
Jul  4 01:45:00.255: %CPPEXMEM-3-NOMEM: F0/0: cpp_cp_svr: QFP: 0, GLOBAL memory allocation
 of 90120448 bytes by FNF failed
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: CPR STILE
EXMEM GRAPH, Allocations: 877, Type: GLOBAL
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: SBC, Bytes
Allocated: 53850112, Type: GLOBAL
```

The warning message does not necessarily indicate a flow monitor application failure. The warning message can indicate internal steps that FNF uses for applying memory from the EXMEM infrastructure.

To ensure that the FNF monitor is enabled successfully, use the **show flow monitor** *monitor-name* command to check **Status** (**allocated** or **not allocated**) of a flow monitor. For more information, see Displaying the Current Status of a Flow Monitor, on page 156.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Configuration Examples for Flexible NetFlow MPLS Egress NetFlow

Example: Configuring Flexible NetFlow Egress Accounting for IPv4 and IPv6 Traffic

The following example shows how to configure Flexible NetFlow egress accounting for IPv4 and IPv6 traffic.

This example starts in global configuration mode.

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exit
!
ip cef
ipv6 cef
!
interface GigabitEthernet0/0/0
 ip address 172.16.6.2 255.255.255.0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ip flow monitor FLOW-MONITOR-1 output
 ipv6 flow monitor FLOW-MONITOR-2 output
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

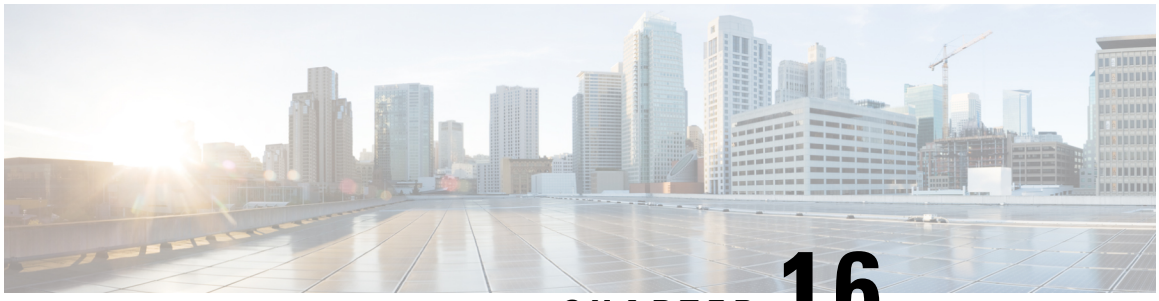
Feature Information for Flexible NetFlow - MPLS Egress NetFlow

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 36: Feature Information for Flexible NetFlow - MPLS Egress NetFlow

Feature Name	Releases	Feature Information
Flexible NetFlow - MPLS Egress NetFlow	12.2(33)SRE 12.2(50)SY 12.4(22)T 15.0(1)SY 15.0(1)SY1 Cisco IOS XE Release 3.1S	The Flexible NetFlow--MPLS Egress NetFlow feature allows you to capture IP flow information for packets undergoing MPLS label disposition; that is, packets that arrive on a router as MPLS packets and are transmitted as IP packets. Support for this feature was added for Cisco 7200 and 7300 NPE series routers in Cisco IOS Release 12.2(33)SRE. No commands were introduced or modified by this feature.



CHAPTER 16

Flexible NetFlow v9 Export Format

This feature enables sending export packets using the Version 9 export format.

Prerequisites for Flexible NetFlow v9 Export Format

- The networking device must be running a Cisco release that supports Flexible NetFlow.

Information About Flexible NetFlow v9 Export Format

Flow Exporters

Flow exporters are created as separate components in a router's configuration. Exporters are assigned to flow monitors to export the data from the flow monitor cache to a remote system such as a NetFlow collector. Flow monitors can support more than one exporter. Each exporter can be customized to meet the requirements of the flow monitor or monitors in which it is used and the NetFlow collector systems to which it is exporting data.

Benefits of Flexible NetFlow Flow Exporters

Flexible NetFlow allows you to configure many different flow exporters, depending on your requirements. Some of the benefits of Flexible NetFlow flow exporters are as follows:

- Using flow exporters, you can create an exporter for every type of traffic that you want to analyze so that you can send each type of traffic to a different NetFlow collector. Original NetFlow sends the data in a cache for all of the analyzed traffic to a maximum of two export destinations.
- Flow exporters support up to ten exporters per flow monitor. Original NetFlow is limited to only two export destinations per cache.

- Flow exporters can use both TCP and UDP for export.
- Depending on your release, flow exporters can use class of service (CoS) in the packets that are sent to export destinations to help ensure that the packets are given the correct priority throughout the network. Original NetFlow exporters do not use CoS in the packets that are sent to export destinations.
- Depending on your release, flow exporter traffic can be encrypted.

How to Configure Flexible NetFlow v9 Export Format

Configuring the Flow Exporter

Perform this required task to configure the flow exporter.

Note Each flow exporter supports only one destination.

You can export to a destination using either an IPv4 or IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
6. **dscp** *dscp*
7. **source** *interface-type interface-number*
8. **output-features**
9. **template data timeout** *seconds*
10. **transport udp** *udp-port*
11. **ttl** *seconds*
12. **end**
13. **show flow exporter** *exporter-name*
14. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates the flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to the datacenter`	(Optional) Configures a description to the exporter that will appear in the configuration and the display of the **show flow exporter** command.
Step 5	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the destination system for the exporter. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**dscp** *dscp* **Example:** `Device(config-flow-exporter)# dscp 63`	(Optional) Configures differentiated services code point (DSCP) parameters for datagrams sent by the exporter. • The range for the *dscp* argument is from 0 to 63. Default: 0.
Step 7	**source** *interface-type interface-number* **Example:** `Device(config-flow-exporter)# source ethernet 0/0`	(Optional) Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 8	**output-features** **Example:** `Device(config-flow-exporter)# output-features`	(Optional) Enables sending export packets using quality of service (QoS) and encryption.
Step 9	**template data timeout** *seconds* **Example:** `Device(config-flow-exporter)# template data timeout 120`	(Optional) Configures resending of templates based on a timeout. • The range for the *seconds* argument is 1 to 86400 (86400 seconds = 24 hours).
Step 10	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 650`	Specifies the UDP port on which the destination system is listening for exported datagrams. • The range for the *udp-port* argument is from 1 to 65536.

	Command or Action	Purpose
Step 11	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 15`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. • The range for the *seconds* argument is from 1 to 255.
Step 12	**end** **Example:** `Device(config-flow-exporter)# end`	Exits flow exporter configuration mode and returns to privileged EXEC mode.
Step 13	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.
Step 14	**show running-config flow exporter** *exporter-name* **Example:** `Device# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Configuration Examples for Flexible NetFlow v9 Export Format

Example: Configuring NetFlow v9 Export Format

The following example shows how to configure version 9 export for Flexible NetFlow.

This example starts in global configuration mode.

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 export-protocol netflow-v9
 transport udp 90
 exit
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exporter EXPORTER-1
!
```

```
ip cef
!
interface GigabitEthernet 0/0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

Additional Reference for Flexible NetFlow v9 Export Format

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards

Standard	Title
None	—

MIBs

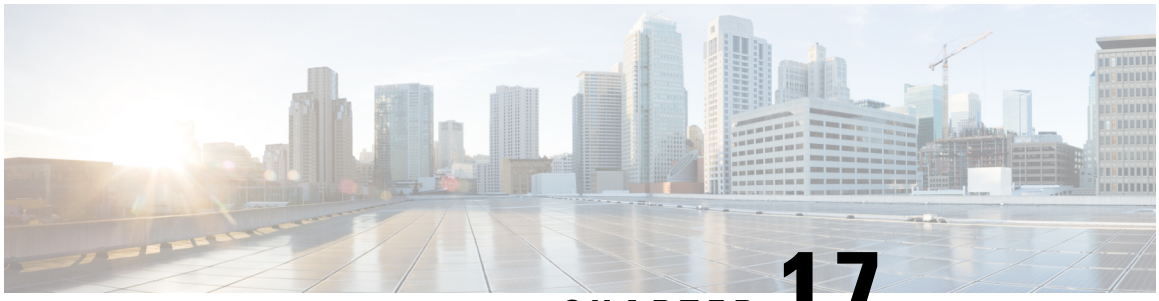
MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3954	*Cisco Systems NetFlow Services Export Version 9*

Technical Assistance

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



CHAPTER 17

Flexible NetFlow Output Features on Data Export

This feature enables sending export packets using Quality of Service (QoS) and encryption.

Prerequisites for Flexible NetFlow Output Features on Data Export

- The networking device must be running a Cisco release that supports Flexible NetFlow.

IPv4 Traffic

- The networking device must be configured for IPv4 routing.
- One of the following must be enabled on your router and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding or distributed Cisco Express Forwarding.

IPv6 Traffic

- The networking device must be configured for IPv6 routing.
- One of the following must be enabled on your router and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding IPv6 or distributed Cisco Express Forwarding.

Information About Flexible NetFlow Output Features on Data Export

Flow Exporters

Flow exporters are created as separate components in a router's configuration. Exporters are assigned to flow monitors to export the data from the flow monitor cache to a remote system such as a NetFlow collector. Flow monitors can support more than one exporter. Each exporter can be customized to meet the requirements of the flow monitor or monitors in which it is used and the NetFlow collector systems to which it is exporting data.

Benefits of Flexible NetFlow Flow Exporters

Flexible NetFlow allows you to configure many different flow exporters, depending on your requirements. Some of the benefits of Flexible NetFlow flow exporters are as follows:

- Using flow exporters, you can create an exporter for every type of traffic that you want to analyze so that you can send each type of traffic to a different NetFlow collector. Original NetFlow sends the data in a cache for all of the analyzed traffic to a maximum of two export destinations.
- Flow exporters support up to ten exporters per flow monitor. Original NetFlow is limited to only two export destinations per cache.
- Flow exporters can use both TCP and UDP for export.
- Depending on your release, flow exporters can use class of service (CoS) in the packets that are sent to export destinations to help ensure that the packets are given the correct priority throughout the network. Original NetFlow exporters do not use CoS in the packets that are sent to export destinations.
- Depending on your release, flow exporter traffic can be encrypted.

How to Configure Flexible NetFlow Output Features on Data Export

Restrictions

Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor. Flow exporters are added to flow monitors to enable data export from the flow monitor cache.

Note Only the keywords and arguments required for the Flexible NetFlow commands used in these tasks are explained in these tasks. For information about the other keywords and arguments available for these Flexible NetFlow commands, refer to the *Cisco IOS Flexible NetFlow Command Reference* .

Configuring the Flow Exporter

Perform this required task to configure the flow exporter.

Note Each flow exporter supports only one destination.

You can export to a destination using either an IPv4 or IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
6. **dscp** *dscp*
7. **source** *interface-type interface-number*
8. **output-features**
9. **template data timeout** *seconds*
10. **transport udp** *udp-port*
11. **ttl** *seconds*
12. **end**
13. **show flow exporter** *exporter-name*
14. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates the flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.

	Command or Action	Purpose
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to the datacenter`	(Optional) Configures a description to the exporter that will appear in the configuration and the display of the **show flow exporter** command.
Step 5	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the destination system for the exporter. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**dscp** *dscp* **Example:** `Device(config-flow-exporter)# dscp 63`	(Optional) Configures differentiated services code point (DSCP) parameters for datagrams sent by the exporter. • The range for the *dscp* argument is from 0 to 63. Default: 0.
Step 7	**source** *interface-type interface-number* **Example:** `Device(config-flow-exporter)# source ethernet 0/0`	(Optional) Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 8	**output-features** **Example:** `Device(config-flow-exporter)# output-features`	(Optional) Enables sending export packets using quality of service (QoS) and encryption.
Step 9	**template data timeout** *seconds* **Example:** `Device(config-flow-exporter)# template data timeout 120`	(Optional) Configures resending of templates based on a timeout. • The range for the *seconds* argument is 1 to 86400 (86400 seconds = 24 hours).
Step 10	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 650`	Specifies the UDP port on which the destination system is listening for exported datagrams. • The range for the *udp-port* argument is from 1 to 65536.
Step 11	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 15`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. • The range for the *seconds* argument is from 1 to 255.
Step 12	**end** **Example:** `Device(config-flow-exporter)# end`	Exits flow exporter configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 13	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.
Step 14	**show running-config flow exporter** *exporter-name* **Example:** `Device# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Displaying the Current Status of a Flow Exporter

To display the current status of a flow exporter, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **show flow exporter** [**export-ids {netflow-v5**| **netflow-v9**} | [**name**] *exporter-name* [**statistics** | **templates**]]

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show flow exporter** [**export-ids {netflow-v5**| **netflow-v9**} | [**name**] *exporter-name* [**statistics** | **templates**]]

The **show flow exporter**command shows the current status of the flow exporter that you specify.

Example:

```
Device# show flow exporter EXPORTER-1
Flow Exporter EXPORTER-1:
  Description:              Exports to the datacenter
  Export protocol:          NetFlow Version 9
  Transport Configuration:
    Destination IP address: 172.16.10.2
    Source IP address:      172.16.6.2
    Source Interface:       GigabitEthernet1/0/0
    Transport Protocol:     UDP
    Destination Port:       650
    Source Port:            55864
    DSCP:                   0x3F
    TTL:                    15
    Output Features:        Used
```

```
Options Configuration:
  exporter-stats (timeout 120 seconds)
  interface-table (timeout 120 seconds)
  sampler-table (timeout 120 seconds)
```

Verifying the Flow Exporter Configuration

To verify the configuration commands that you entered, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

Step 1 enable

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 show running-config flow exporter *exporter-name*

The **show running-config flow exporter** command shows the configuration commands of the flow exporter that you specify.

Example:

```
Device# show running-config flow exporter EXPORTER-1
Building configuration...
Current configuration:
!
flow exporter EXPORTER-1
 description Exports to the datacenter
 destination 172.16.10.2
 source GigabitEthernet1/0/0
 dscp 63
 ttl 15
 transport udp 650
 template data timeout 120
 option exporter-stats timeout 120
 option interface-table timeout 120
 option sampler-table timeout 120
!
end
```

Configuring and Enabling Flexible NetFlow with Data Export

You must create a flow monitor to configure the types of traffic for which you want to export the cache data. You must enable the flow monitor by applying it to at least one interface to start exporting data. To configure and enable Flexible NetFlow with data export, perform this required task.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. The record format can be one of the predefined record formats, or an advanced user may create his or her own record format using the **collect** and **match**commands in Flexible NetFlow flow record configuration mode.

Note You must remove a flow monitor from all of the interfaces to which you have applied it before you can modify the **record** format of the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6** *record* [**peer**] }]
5. **exporter** *exporter-name*
6. **exit**
7. **interface** *type number*
8. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.

	Command or Action	Purpose
Step 4	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6** *record* [**peer**] }] **Example:** `Device(config-flow-monitor)# record netflow ipv4 original-input`	Specifies the record for the flow monitor.
Step 5	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	Specifies the name of an exporter that you created previously.
Step 6	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 8	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates the flow monitor that you created previously by assigning it to the interface to analyze traffic.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]][**statistics**]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status and statistics for a Flexible NetFlow flow monitor. This will verify data export is enabled for the flow monitor cache.

Configuration Examples for Flexible NetFlow Output Features on Data Export

Example: Configuring Sending Export Packets Using QoS

The following example shows how to enable QoS on Flexible Netflow export packets.

Note The Flexible NetFlow export packets are transmitted using QoS on Ethernet interface 0/1 (the interface on which the destination is reachable) to the destination host (IP address 10.0.1.2).

This sample starts in global configuration mode:

```
!
flow record FLOW-RECORD-1
 match ipv4 source address
 collect counter packets
!
flow exporter FLOW-EXPORTER-1
 destination 10.0.1.2
 output-features
 dscp 18
!
flow monitor FLOW-MONITOR-1
 record FLOW-RECORD-1
 exporter FLOW-EXPORTER-1
 cache entries 1024
!
ip cef
!
class-map match-any COS3
!
policy-map PH_LABS_FRL_64k_16k_16k_8k_8k
 class COS3
  bandwidth percent 2
  random-detect dscp-based
  random-detect exponential-weighting-constant 1
  random-detect dscp 18 200 300 10
!
interface Ethernet 0/0
 ip address 10.0.0.1 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
interface Ethernet 0/1
 ip address 10.0.1.1 255.255.255.0
 service-policy output PH_LABS_FRL_64k_16k_16k_8k_8k
!
```

The following display output shows that the flow monitor is exporting data using output feature support that enables the exported data to use QoS:

```
Device# show flow monitor FLOW-MONITOR-1
Flow Exporter FLOW-EXPORTER-1:
  Description:              User defined
```

```
Tranport Configuration:
  Destination IP address: 10.0.1.2
  Source IP address:      10.0.0.1
  Transport Protocol:     UDP
  Destination Port:       9995
  Source Port:            56750
  DSCP:                   0x12
  TTL:                    255
  Output Features:        Used
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

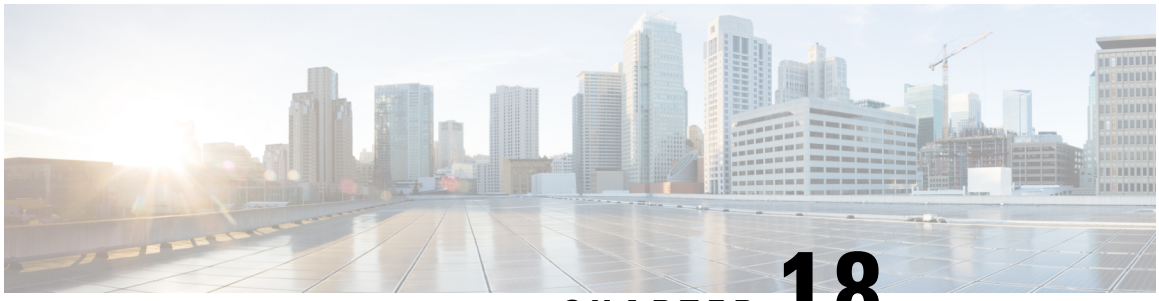
Feature Information for Flexible NetFlow—Output Features on Data Export

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 37: Feature Information for Flexible NetFlow—Output Features on Data Export

Feature Name	Releases	Feature Information
Flexible NetFlow—Output Features on Data Export	12.4(20)T Cisco IOS XE Release 3.1S	Enables sending export packets using QoS and encryption. The following command was introduced: **output-features**.



CHAPTER 18

Flexible NetFlow NetFlow V5 Export Protocol

The Flexible Netflow NetFlow V5 Export Protocol feature enables sending export packets using the Version 5 export protocol.

Support for this feature was added for Cisco 7200 and 7300 Network Processing Engine (NPE) series routers in Cisco IOS Release 12.2(33)SRE.

Restrictions for Flexible NetFlow NetFlow V5 Export Protocol

- The NetFlow Version 5 export protocol that was first shipped in Cisco IOS Release 12.4(22)T is supported for flow monitors that use only the following Flexible NetFlow predefined records: netflow-original, original input, and original output.

Information about Flexible NetFlow NetFlow V5 Export Protocol

Flexible NetFlow V5 Export Protocol Overview

This feature enables sending export packets using the Version 5 export protocol.

How to Configure Flexible NetFlow NetFlow V5 Export Protocol

Configuring the Flow Exporter

Perform this required task to configure the flow exporter.

Note Each flow exporter supports only one destination.

You can export to a destination using either an IPv4 or IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
6. **dscp** *dscp*
7. **source** *interface-type interface-number*
8. **output-features**
9. **template data timeout** *seconds*
10. **transport udp** *udp-port*
11. **ttl** *seconds*
12. **end**
13. **show flow exporter** *exporter-name*
14. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates the flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to the datacenter`	(Optional) Configures a description to the exporter that will appear in the configuration and the display of the **show flow exporter** command.

	Command or Action	Purpose
Step 5	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the destination system for the exporter. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**dscp** *dscp* **Example:** `Device(config-flow-exporter)# dscp 63`	(Optional) Configures differentiated services code point (DSCP) parameters for datagrams sent by the exporter. • The range for the *dscp* argument is from 0 to 63. Default: 0.
Step 7	**source** *interface-type interface-number* **Example:** `Device(config-flow-exporter)# source ethernet 0/0`	(Optional) Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 8	**output-features** **Example:** `Device(config-flow-exporter)# output-features`	(Optional) Enables sending export packets using quality of service (QoS) and encryption.
Step 9	**template data timeout** *seconds* **Example:** `Device(config-flow-exporter)# template data timeout 120`	(Optional) Configures resending of templates based on a timeout. • The range for the *seconds* argument is 1 to 86400 (86400 seconds = 24 hours).
Step 10	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 650`	Specifies the UDP port on which the destination system is listening for exported datagrams. • The range for the *udp-port* argument is from 1 to 65536.
Step 11	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 15`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. • The range for the *seconds* argument is from 1 to 255.
Step 12	**end** **Example:** `Device(config-flow-exporter)# end`	Exits flow exporter configuration mode and returns to privileged EXEC mode.
Step 13	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.

	Command or Action	Purpose
Step 14	**show running-config flow exporter** *exporter-name* **Example:** `Device# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Configuration Examples for Flexible NetFlow NetFlow V5 Export Protocol

Example: Configuring Version 5 Export

The following example shows how to configure version 5 export for Flexible NetFlow.

This sample starts in global configuration mode:

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 export-protocol netflow-v5
 transport udp 90
 exit
!
flow monitor FLOW-MONITOR-1
 record netflow ipv4 original-input
 exporter EXPORTER-1
!
ip cef
!
interface Ethernet 0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

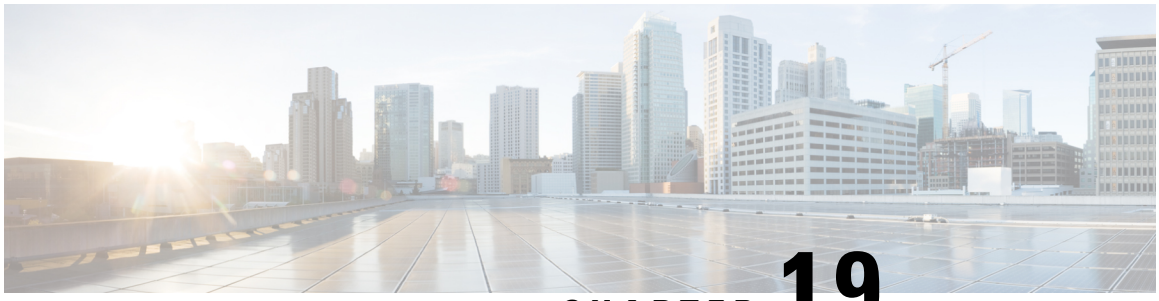
Feature Information for Flexible NetFlow NetFlow V5 Export Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 38: Feature Information for Flexible NetFlow NetFlow V5 Export Protocol

Feature Name	Releases	Feature Information
Flexible NetFlow--NetFlow V5 Export Protocol	12.2(33)SRE 12.2(50)SY 12.4(22)T 15.0(1)SY 15.0(1)SY1 Cisco IOS XE Release 3.1S	Enables sending export packets using the Version 5 export protocol. Support for this feature was added for Cisco 7200 and 7300 Network Processing Engine (NPE) series routers in Cisco IOS Release 12.2(33)SRE. The following command was introduced: **export-protocol**.



CHAPTER 19

Using Flexible NetFlow Flow Sampling

This document contains information about and instructions for configuring sampling to reduce the CPU overhead of analyzing traffic with Flexible NetFlow.

NetFlow is a Cisco technology that provides statistics on packets flowing through a router. NetFlow is the standard for acquiring IP operational data from IP networks. NetFlow provides data to support network and security monitoring, network planning, traffic analysis, and IP accounting.

Flexible NetFlow improves on original NetFlow by adding the capability to customize the traffic analysis parameters for your specific requirements. Flexible NetFlow facilitates the creation of more complex configurations for traffic analysis and data export through the use of reusable configuration components.

Prerequisites for Using Flexible NetFlow Flow Sampling

- The networking device must be running a Cisco release that supports Flexible NetFlow.

Restrictions for Using Flexible NetFlow Flow Sampling

Information About Flexible NetFlow Flow Sampling

Flow Samplers

Flow samplers are created as separate components in a router's configuration. Flow samplers are used to reduce the load on the device that is running by limiting the number of packets that are selected for analysis.

Flow sampling exchanges monitoring accuracy for router performance. When you apply a sampler to a flow monitor, the overhead load on the router of running the flow monitor is reduced because the number of packets that the flow monitor must analyze is reduced. The reduction in the number of packets that are analyzed by the flow monitor causes a corresponding reduction in the accuracy of the information stored in the flow monitor's cache.

Samplers are combined with flow monitors when they are applied to an interface with the **ip flow monitor** command.

How to Configure Flexible NetFlow Flow Sampling

Flow sampling reduces the CPU overhead of analyzing traffic with Flexible NetFlow by reducing the number of packets that are analyzed.

Note Only the keywords and arguments required for the Flexible NetFlow commands used in these tasks are explained in these tasks. For information about the other keywords and arguments available for these Flexible NetFlow commands, refer to the *Cisco IOS Flexible NetFlow Command Reference* .

Configuring a Flow Monitor

Samplers are applied to an interface in conjunction with a flow monitor. You must create a flow monitor to configure the types of traffic that you want to analyze before you can enable sampling. Perform this required task to configure a flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. The record format can be one of the predefined record formats, or an advanced user may create his or her own record format using the **collect** and **match** commands in Flexible NetFlow flow record configuration mode.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]} **Example:** `Device(config-flow-monitor)# record netflow ipv4 original-input`	Specifies the record for the flow monitor.
Step 6	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.

Perform this required task to configure and enable a flow sampler.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sampler** *sampler-name*
4. **description** *description*
5. **mode** {**random**} **1 out-of** *window-size*
6. **exit**
7. **interface** *type number*

8. {**ip** | **ipv6**} **flow monitor** *monitor-name* [[**sampler**] *sampler-name*] {**input** | **output**}
9. **end**
10. **show sampler sampler-name**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**sampler** *sampler-name* **Example:** `Device(config)# sampler SAMPLER-1`	Creates a sampler and enters sampler configuration mode. • This command also allows you to modify an existing sampler.
Step 4	**description** *description* **Example:** `Device(config-sampler)# description Sample at 50%`	(Optional) Creates a description for the flow sampler.
Step 5	**mode** {**random**} **1 out-of** *window-size* **Example:** `Device(config-sampler)# mode random 1 out-of 2`	Specifies the sampler mode and the flow sampler window size. • The range for the *window-size* argument is from .
Step 6	**exit** **Example:** `Device(config-sampler)# exit`	Exits sampler configuration mode and returns to global configuration mode.
Step 7	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 8	{**ip**	**ipv6**} **flow monitor** *monitor-name* [[**sampler**] *sampler-name*] {**input**	**output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input`	Assigns the flow monitor and the flow sampler that you created to the interface to enable sampling.

	Command or Action	Purpose
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	**show sampler sampler-name** **Example:** `Device# show sampler SAMPLER-1`	Displays the status and statistics of the flow sampler that you configured and enabled.

Displaying the Status and Statistics of the Flow Sampler Configuration

To display the status and statistics of the flow sampler that you configured and enabled, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **show sampler sampler-name**

DETAILED STEPS

Step 1 **enable**

The **enable** command enters privileged EXEC mode (enter the password if prompted).

Example:

```
Device> enable
Device#
```

Step 2 **show sampler sampler-name**

The **show sampler** command shows the current status of the sampler that you specify.

Example:

```
Device# show sampler SAMPLER-1
Sampler SAMPLER-1:
  ID:             2
  Description:    Sample at 50%
  Type:           random
  Rate:           1 out of 2
  Samples:        2482
  Requests:       4964
  Users (1):
    flow monitor FLOW-MONITOR-1 (ip,Et0/0,I  2482 out of 4964
```

Configuration Examples for Flexible NetFlow Flow Sampling

Example: Configuring and Enabling a Random Sampler for IPv4 Traffic

The following example shows how to configure and enable random sampling for IPv4 output traffic.

This example starts in global configuration mode.

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
sampler SAMPLER-1
 mode random 1 out-of 2
 exit
!
ip cef
!
interface GigabitEthernet 0/0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 output
!
```

The following example shows how to configure and enable random sampling for IPv4 input traffic.

This example starts in global configuration mode.

```
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!
flow monitor FLOW-MONITOR-1
 record v4_r1
 exit
!
sampler SAMPLER-1
 mode random 1 out-of 2
 exit
!
ip cef
!
```

```
interface GigabitEthernet 0/0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input
!
```

Example: Adding a Sampler to a Flow Monitor When a Flow Monitor Is Already Enabled

The following example shows what happens when you try to add a sampler to a flow monitor that has already been enabled on an interface without a sampler:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-2 input
% Flow Monitor: Flow Monitor 'FLOW-MONITOR-1' is already on in full mode and cannot be
enabled with a sampler.
```

The following example shows how to remove the flow monitor from the interface so that it can be enabled with the sampler:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# no ip flow monitor FLOW-MONITOR-1 input
Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-2 input
```

Example: Removing a Sampler from a Flow Monitor

The following example shows what happens when you try to remove a sampler from a flow monitor on an interface by entering the **ip flow monitor** command again without the sampler keyword and argument:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip flow monitor FLOW-MONITOR-1 input
% Flow Monitor: Flow Monitor 'FLOW-MONITOR-1' is already on in sampled mode and cannot be
enabled in full mode.
```

The following example shows how to remove the flow monitor that was enabled with a sampler from the interface so that it can be enabled without the sampler:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# no ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-2 input
Device(config-if)# ip flow monitor FLOW-MONITOR-1 input
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*

Related Topic	Document Title
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

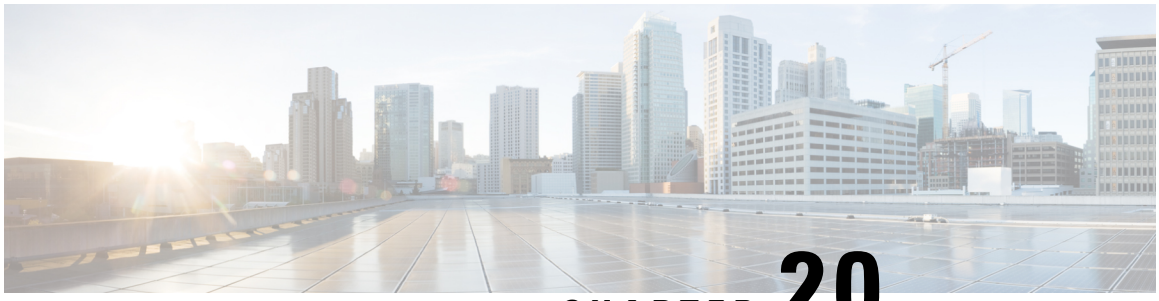
Feature Information for Flexible NetFlow Flow Sampling

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 39: Feature Information for Flexible Netflow Flow Sampling

Feature Name	Releases	Feature Information
Flexible Netflow - Random Sampling	12.2(50)SY 12.4(20)T Cisco IOS XE Release 3.1S Cisco IOS XE Release 3.2SE	Flow samplers are created as separate components in a router's configuration. Flow samplers are used to reduce the load on the device that is running Flexible NetFlow by limiting the number of packets that are selected for analysis. Samplers use either random or deterministic sampling techniques (modes). The following commands were introduced or modified: **clear sampler**, **debug sampler** , **mode** , **record** , **sampler** , **show sampler** .



CHAPTER 20

Flexible NetFlow - Layer 2 Fields

The Flexible NetFlow - Layer 2 Fields feature enables collecting statistics for Layer 2 fields such as MAC addresses and virtual LAN (VLAN) IDs from traffic.

Restrictions for Flexible Netflow - Layer 2

- Flexible NetFlow is not supported on L2 interface.

Information About Flexible NetFlow Layer 2 Fields

Flexible NetFlow - Layer 2 Fields Overview

The Flexible NetFlow - Layer 2 Fields feature enables collecting statistics for Layer 2 fields such as MAC addresses and virtual LAN (VLAN) IDs from traffic.

How to Configure Flexible NetFlow Layer 2 Fields

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ip** | **ipv6**} {**destination** | **source**} **address**
6. Repeat Step 5 as required to configure additional key fields for the record.
7. **match flow cts** {**source** | **destination**} **group-tag**
8.
9. Repeat the above step as required to configure additional nonkey fields for the record.
10. **end**
11. **show flow record** *record-name*
12. **show running-config flow record** *record-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record FLOW-RECORD-1`	Creates a flow record and enters Flexible NetFlow flow record configuration mode. • This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.

	Command or Action	Purpose
Step 5	**match** {**ip** | **ipv6**} {**destination** | **source**} **address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	**Note** This example configures the IPv4 destination address as a key field for the record. For information about the other key fields available for the **match ipv4** command, and the other **match** commands that are available to configure key fields.
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—

Step 7	**match flow cts** {**source** | **destination**} **group-tag**
	Example:
	`Device(config-flow-record)# match flow cts source group-tag`
	`Device(config-flow-record)# match flow cts destination group-tag`
		Note This example configures the CTS source group tag and destination group tag as a key field for the record. For information about the other key fields available for the **match ipv4**/**ipv6** command, and the other **match** commands that are available to configure key fields.
		Note
		• Ingress:
		• In an incoming packet, if a header is present, SGT will reflect the same value as the header. If no value is present, it will show zero.
		• The DGT value will not depend on the ingress port SGACL configuration.
		• Egress:
		• If either propagate SGT or CTS is disabled on the egress interface, then SGT will be zero.
		• In an outgoing packet, if SGACL configuration that corresponds to the (SGT, DGT) exists, DGT will be non-zero.
		• If SGACL is disabled on the egress port/VLAN or if global SGACL enforcement is disabled, then DGT will be zero

Step 8	**Example:**	Configures the input interface as a nonkey field for the record. **Note** This example configures the input interface as a nonkey field for the record.

	Command or Action	Purpose
Step 9	Repeat the above step as required to configure additional nonkey fields for the record.	—
Step 10	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.
Step 11	**show flow record** *record-name* **Example:** `Device# show flow record FLOW_RECORD-1`	(Optional) Displays the current status of the specified flow record.
Step 12	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*

9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name*} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active**} *seconds* | { **normal** } **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 9	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status for a Flexible NetFlow flow monitor.
Step 11	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

While running the **ip flow monitor** command for the first interface to enable FNF monitor, you may see the following warning message displaying a GLOBAL memory allocation failure. This log is triggered by enabling FNF monitoring with a large cache size.

```
Jul  4 01:45:00.255: %CPPEXMEM-3-NOMEM: F0/0: cpp_cp_svr: QFP: 0, GLOBAL memory allocation
 of 90120448 bytes by FNF failed
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: CPR STILE
EXMEM GRAPH, Allocations: 877, Type: GLOBAL
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: SBC, Bytes
Allocated: 53850112, Type: GLOBAL
```

The warning message does not necessarily indicate a flow monitor application failure. The warning message can indicate internal steps that FNF uses for applying memory from the EXMEM infrastructure.

To ensure that the FNF monitor is enabled successfully, use the **show flow monitor** *monitor-name* command to check **Status** (**allocated** or **not allocated**) of a flow monitor. For more information, see Displaying the Current Status of a Flow Monitor, on page 156.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Configuration Examples for Flexible NetFlow Layer 2 Fields

Example: Configuring Flexible NetFlow for Monitoring MAC and VLAN Statistics

The following example shows how to configure Flexible NetFlow for monitoring MAC and VLAN statistics.

This example starts in global configuration mode.

```
!
 flow record LAYER-2-FIELDS-1
 match ipv4 source address
 match ipv4 destination address
 match datalink dot1q vlan output
 match datalink mac source address input
 match datalink mac source address output
 match datalink mac destination address input
 match flow direction
!
 exit
!
!
flow monitor FLOW-MONITOR-4
 record LAYER-2-FIELDS-1
 exit
!
ip cef
!
interface GigabitEthernet0/0/1
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

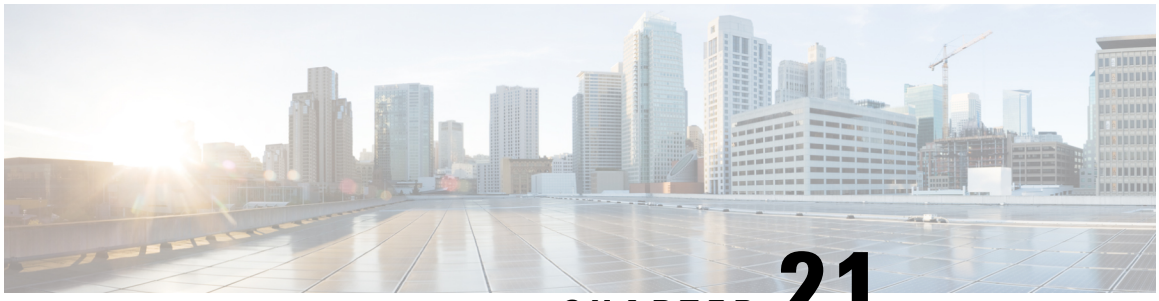
Feature Information for Flexible NetFlow - Layer 2 Fields

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 40: Feature Information for Flexible NetFlow - Layer 2 Fields

Feature Name	Releases	Feature Information
Flexible NetFlow - Layer 2 Fields	12.2(33)SRE 12.4(22)T Cisco IOS XE Release 3.2SE	Enables collecting statistics for Layer 2 fields such as MAC addresses and virtual LAN (VLAN) IDs from traffic. Support for this feature was added for Cisco 7200 and 7300 Network Processing Engine (NPE) series routers in Cisco IOS Release 12.2(33)SRE. The following commands were introduced or modified: **collect datalink dot1q vlan**, **collect datalink mac**, **match datalink dot1q vlan**, **match datalink mac**.



CHAPTER 21

Flexible Netflow - Ingress VRF Support

The Flexible Netflow - Ingress VRF Support feature enables collecting the virtual routing and forwarding (VRF) ID from incoming packets on a router by applying an input flow monitor having a flow record that collects the VRF ID as a key or a nonkey field.

Information About Flexible NetFlow Ingress VRF Support

Flexible NetFlow—Ingress VRF Support Overview

This feature enables collecting the virtual routing and forwarding (VRF) ID from incoming packets on a router by applying an input flow monitor having a flow record that collects the VRF ID as a key or a nonkey field.

How to Configure Flexible NetFlow Ingress VRF Support

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ip** | **ipv6**} {**destination** | **source**} **address**
6. Repeat Step 5 as required to configure additional key fields for the record.
7. **match flow cts** {**source** | **destination**} **group-tag**
8.
9. Repeat the above step as required to configure additional nonkey fields for the record.
10. **end**
11. **show flow record** *record-name*
12. **show running-config flow record** *record-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record FLOW-RECORD-1`	Creates a flow record and enters Flexible NetFlow flow record configuration mode. • This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.
Step 5	**match** {**ip**	**ipv6**} {**destination**	**source**} **address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	**Note** This example configures the IPv4 destination address as a key field for the record. For information about the other key fields available for the **match ipv4** command, and the other **match** commands that are available to configure key fields.
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—

	Command or Action	Purpose

Step 7	**match flow cts** {**source** | **destination**} **group-tag**
	Example:
	`Device(config-flow-record)# match flow cts source group-tag`
	`Device(config-flow-record)# match flow cts destination group-tag`
		Note This example configures the CTS source group tag and destination group tag as a key field for the record. For information about the other key fields available for the **match ipv4**/**ipv6** command, and the other **match** commands that are available to configure key fields.
		Note
		• Ingress:
		• In an incoming packet, if a header is present, SGT will reflect the same value as the header. If no value is present, it will show zero.
		• The DGT value will not depend on the ingress port SGACL configuration.
		• Egress:
		• If either propagate SGT or CTS is disabled on the egress interface, then SGT will be zero.
		• In an outgoing packet, if SGACL configuration that corresponds to the (SGT, DGT) exists, DGT will be non-zero.
		• If SGACL is disabled on the egress port/VLAN or if global SGACL enforcement is disabled, then DGT will be zero

Step 8	**Example:**	Configures the input interface as a nonkey field for the record. **Note** This example configures the input interface as a nonkey field for the record.
Step 9	Repeat the above step as required to configure additional nonkey fields for the record.	—
Step 10	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.
Step 11	**show flow record** *record-name* **Example:**	(Optional) Displays the current status of the specified flow record.

	Command or Action	Purpose
	`Device# show flow record FLOW_RECORD-1`	
Step 12	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name*} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active**} *seconds* | { **normal** } **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 9	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status for a Flexible NetFlow flow monitor.

	Command or Action	Purpose
Step 11	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

While running the **ip flow monitor** command for the first interface to enable FNF monitor, you may see the following warning message displaying a GLOBAL memory allocation failure. This log is triggered by enabling FNF monitoring with a large cache size.

```
Jul  4 01:45:00.255: %CPPEXMEM-3-NOMEM: F0/0: cpp_cp_svr: QFP: 0, GLOBAL memory allocation
 of 90120448 bytes by FNF failed
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: CPR STILE
EXMEM GRAPH, Allocations: 877, Type: GLOBAL
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: SBC, Bytes
Allocated: 53850112, Type: GLOBAL
```

The warning message does not necessarily indicate a flow monitor application failure. The warning message can indicate internal steps that FNF uses for applying memory from the EXMEM infrastructure.

To ensure that the FNF monitor is enabled successfully, use the **show flow monitor** *monitor-name* command to check **Status** (**allocated** or **not allocated**) of a flow monitor. For more information, see Displaying the Current Status of a Flow Monitor, on page 156.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Configuration Examples for Flexible NetFlow Ingress VRF Support

Example: Configuring Flexible NetFlow for Ingress VRF Support

The following example configures the collection of the virtual routing and forwarding (VRF) ID from incoming packets on a router by applying an input flow monitor having a flow record that collects the VRF ID as a key field.

This example starts in global configuration mode.

```
!
flow record rm_1
match routing vrf input
match ipv4 source address
match ipv4 destination address
collect interface input
collect interface output
collect counter packets
!
flow monitor mm_1
record rm_1
!
interface GigabitEthernet 0/0/0
ip vrf forwarding green
ip address 172.16.2.2 255.255.255.252
ip flow monitor mm_1 input
!
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

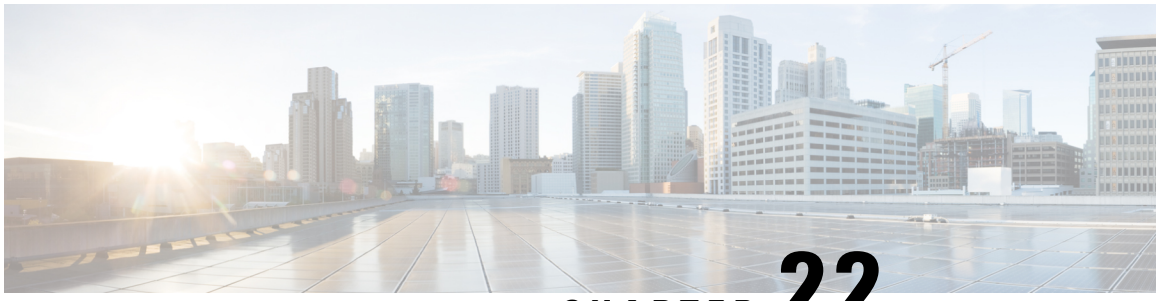
Feature Information for Flexible NetFlow—Ingress VRF Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 41: Feature Information for Flexible NetFlow—Ingress VRF Support

Feature Name	Releases	Feature Information
Flexible NetFlow--Ingress VRF Support	12.2(33)SRE 12.2(50)SY 15.0(1)M 15.0(1)SY 15.0(1)SY1	Enables collecting the virtual routing and forwarding (VRF) ID from incoming packets on a router by applying an input flow monitor having a flow record that collects the VRF ID as a key or a nonkey field. Support for this feature was added for Cisco 7200 and 7300 Network Processing Engine (NPE) series routers in Cisco IOS Release 12.2(33)SRE. The following commands were introduced or modified: **collect routing, match routing, option (Flexible NetFlow), show flow monitor.**



CHAPTER 22

Flexible NetFlow NBAR Application Recognition Overview

NBAR enables creation of different flows for each application seen between any two IP hosts by applying a flow monitor having a flow record that collects the application name as a key or a nonkey field.

Information About Flexible NetFlow NBAR Application Recognition

Flexible NetFlow NBAR Application Recognition Overview

If the predefined Flexible NetFlow records are not suitable for your traffic requirements, you can create a user-defined (custom) record using the Flexible NetFlow **collect** and **match** commands. Before you can create a customized record, you must decide the criteria that you are going to use for the key and nonkey fields.

If you want to create a customized record for detecting network attacks, you must include the appropriate key and nonkey fields in the record to ensure that the router creates the flows and captures the data that you need to analyze the attack and respond to it. Flexible NetFlow uses Network-based Application recognition (NBAR) to create different flows for each application seen between any two IP hosts by applying a flow monitor having a flow record that collects the application name as a key field.

How to Configure Flexible NetFlow NBAR Application Recognition

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ip** | **ipv6**} {**destination** | **source**} **address**
6. Repeat Step 5 as required to configure additional key fields for the record.
7. **match flow cts** {**source** | **destination**} **group-tag**
8.
9. Repeat the above step as required to configure additional nonkey fields for the record.
10. **end**
11. **show flow record** *record-name*
12. **show running-config flow record** *record-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:**	Creates a flow record and enters Flexible NetFlow flow record configuration mode.

	Command or Action	Purpose
	`Device(config)# flow record FLOW-RECORD-1`	• This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.
Step 5	**match** {**ip** | **ipv6**} {**destination** | **source**} **address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	**Note** This example configures the IPv4 destination address as a key field for the record. For information about the other key fields available for the **match ipv4** command, and the other **match** commands that are available to configure key fields.
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—

Step 7	**match flow cts** {**source** | **destination**} **group-tag**
	Example:
	`Device(config-flow-record)# match flow cts source group-tag`
	`Device(config-flow-record)# match flow cts destination group-tag`
		Note This example configures the CTS source group tag and destination group tag as a key field for the record. For information about the other key fields available for the **match ipv4**/**ipv6** command, and the other **match** commands that are available to configure key fields.
		Note
		• Ingress:
		• In an incoming packet, if a header is present, SGT will reflect the same value as the header. If no value is present, it will show zero.
		• The DGT value will not depend on the ingress port SGACL configuration.
		• Egress:
		• If either propagate SGT or CTS is disabled on the egress interface, then SGT will be zero.
		• In an outgoing packet, if SGACL configuration that corresponds to the (SGT, DGT) exists, DGT will be non-zero.
		• If SGACL is disabled on the egress port/VLAN or if global SGACL enforcement is disabled, then DGT will be zero

	Command or Action	Purpose
Step 8	**Example:**	Configures the input interface as a nonkey field for the record. **Note** This example configures the input interface as a nonkey field for the record.
Step 9	Repeat the above step as required to configure additional nonkey fields for the record.	—
Step 10	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.
Step 11	**show flow record** *record-name* **Example:** `Device# show flow record FLOW_RECORD-1`	(Optional) Displays the current status of the specified flow record.
Step 12	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*

4. **description** *description*
5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name*} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active**} *seconds*	{ **normal** } **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:**	(Optional) Specifies the name of an exporter that was created previously.

	Command or Action	Purpose
	`Device(config-flow-monitor)# exporter EXPORTER-1`	
Step 9	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status for a Flexible NetFlow flow monitor.
Step 11	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

While running the **ip flow monitor** command for the first interface to enable FNF monitor, you may see the following warning message displaying a GLOBAL memory allocation failure. This log is triggered by enabling FNF monitoring with a large cache size.

```
Jul  4 01:45:00.255: %CPPEXMEM-3-NOMEM: F0/0: cpp_cp_svr: QFP: 0, GLOBAL memory allocation
 of 90120448 bytes by FNF failed
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: CPR STILE
EXMEM GRAPH, Allocations: 877, Type: GLOBAL
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: SBC, Bytes
Allocated: 53850112, Type: GLOBAL
```

The warning message does not necessarily indicate a flow monitor application failure. The warning message can indicate internal steps that FNF uses for applying memory from the EXMEM infrastructure.

To ensure that the FNF monitor is enabled successfully, use the **show flow monitor** *monitor-name* command to check **Status** (**allocated** or **not allocated**) of a flow monitor. For more information, see Displaying the Current Status of a Flow Monitor, on page 156.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.

6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Configuration Examples for Flexible NetFlow NBAR Application Recognition

Example: Configuring Flexible NetFlow for Network-Based Application Recognition

```
!
flow record rm_1
match application name
match ipv4 source address
match ipv4 destination address
collect interface input
collect interface output
collect counter packets
!
flow monitor mm_1
record rm_1
!
interface FastEthernet0/0
ip address 172.16.2.2 255.255.255.0
ip flow monitor mm_1 input
!
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

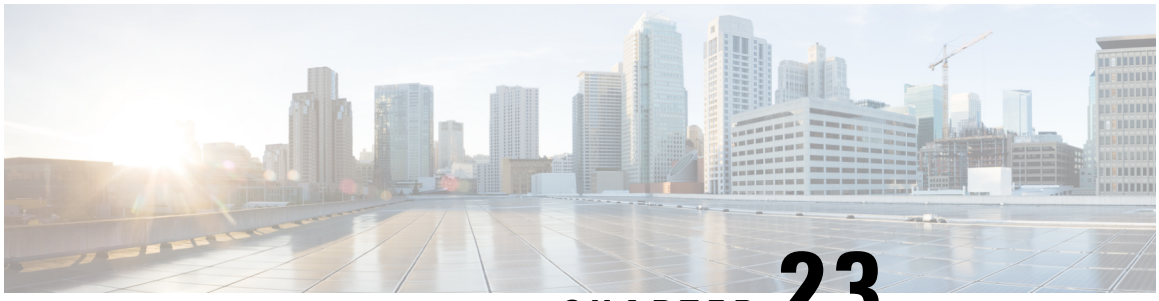
Feature Information for Flexible NetFlow NBAR Application Recognition

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 42: Feature Information for Flexible NetFlow NBAR Application Recognition

Feature Name	Releases	Feature Information
Flexible NetFlow--NBAR Application Recognition		Network-based Application recognition (NBAR) enables creation of different flows for each application seen between any two IP hosts by applying a flow monitor having a flow record that collects the application name as a key or a nonkey field. The following commands were introduced or modified: **collect application name**, **match application name**, **option** (Flexible NetFlow), **show flow monitor**.



CHAPTER 23

Support for ISSU and SSO

High Availability (HA) support for Flexible Netflow is introduced by providing support for both In-Service Software Upgrade (ISSU) and Stateful Switchover (SSO).

These features are enabled by default when the redundancy mode of operation is set to SSO.

Prerequisites for Flexible Netflow High Availability

- The Cisco ISSU process must be configured and working properly. See the “Cisco In-Service Software Upgrade Process” feature module for more information.
- SSO must be configured and working properly. See the “Stateful Switchover” feature module for more information.
- Nonstop Forwarding (NSF) must be configured and working properly. See the “Cisco Nonstop Forwarding” feature module for more information.

Information About Flexible Netflow High Availability

ISSU

The ISSU process allows Cisco software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco software to be modified while packet forwarding continues, which increases network availability and reduces downtime caused by planned software upgrades.

SSO

SSO refers to the implementation of Cisco software that allows applications and features to maintain a defined state between an active and standby Route Processor (RP).

In specific Cisco networking devices that support dual RPs, SSO takes advantage of RP redundancy to increase network availability. The SSO feature takes advantage of RP redundancy by establishing one of the RPs as the active RP while the other RP is designated as the standby RP, and then synchronizing critical state information between them. Following an initial synchronization between the two processors, SSO dynamically maintains RP state information between them.

A switchover from the active to the standby processor occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance.

How to Configure Flexible Netflow High Availability

There are no configuration tasks specific to Flexible Netflow.

The Flexible Netflow high availability features are enabled by default when the redundancy mode of operation is set to SSO.

How to Verify Flexible Netflow High Availability

SUMMARY STEPS

1. **enable**
2. **show redundancy** [**clients** | **counters** | **history** | **switchover history** | **states**
3. **show redundancy states**
4. **show sampler broker** [**detail**] | [**picture**]
5. **show flow exporter broker** [**detail**] | [**picture**]
6. **show flow record broker** [**detail**] | [**picture**]
7. **show flow monitor broker** [**detail**] | [**picture**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show redundancy** [**clients** | **counters** | **history** | **switchover history** | **states** **Example:** `Device# show redundancy`	Displays SSO configuration information.

	Command or Action	Purpose
Step 3	**show redundancy states** **Example:** `Device# show redundancy states`	Verifies that the device is running in SSO mode.
Step 4	**show sampler broker** [**detail**] | [**picture**] **Example:** `Device# show sampler broker detail`	Displays information about the state of the exporter broker for the Flexible Netflow sampler.
Step 5	**show flow exporter broker** [**detail**] | [**picture**] **Example:** `Device# show flow exporter broker detail`	Displays information about the state of the broker for the Flexible Netflow flow exporter.
Step 6	**show flow record broker** [**detail**] | [**picture**] **Example:** `Device# show flow record broker detail`	Displays information about the state of the broker for the Flexible Netflow flow record.
Step 7	**show flow monitor broker** [**detail**] | [**picture**] **Example:** `Device# show flow monitor broker detail`	Displays information about the state of the broker for the Flexible Netflow flow monitor.

What to do next

Configuration Examples for Flexible Netflow High Availability

There are no configuration examples for Flexible Netflow high availability features.

All examples are for displaying the status of Flexible Netflow high availability.

Example: Displaying Detailed Status for the Sampler Broker

The following example shows the status output for the Flexible Netflow flow record broker. This output is very similar to the output for the other Flexible Netflow brokers: the sampler broker, the flow exporter broker, and the flow monitor broker.

```
Device# show flow record broker detail
Brokering for Linecard 7 (0x80)
Multicast groups :-
 0x7F801C95D000
 Linecard 7 (0x80) enabled for download
 Consume report for Linecard 7 (0x80) (pos 1)
 24/0 completed/pending updates (all VRFs)
Update list ranges from pos 1 to pos 0 :-
```

```
1 - 24 updates
0 - 0 updates
Broker records :-
* -   - Start of list
1 -   - Flush
1 - Mod - Create netflow-v5
1 - Mod - Create options interface-table
1 - Mod - Create options exporter-statistics
1 - Mod - Create options vrf-id-name-table
1 - Mod - Create options sampler-table
1 - Mod - Create options applications-name
1 - Mod - Create netflow-original
1 - Mod - Create netflow ipv4 original-input
```

Example: Displaying a Status Summary for the Flow Record Broker

The following example shows a status summary output for the Flexible Netflow flow record broker. This output is very similar to the output for the other Flexible Netflow brokers: the sampler broker, the flow exporter broker, and the flow monitor broker.

```
Device# show flow record broker picture
Key:
 '['=start record, ']'=end record, 'F'=flush record, 'D'=display record
 '+<n>'=sequenve of <n> Modify update records
 '-<n>'=sequenve of <n> Delete update records
 'C<<lc>:<vrf>>'=consume record for linecard(s) <lc> and VRF(s) <vrf> <*=all>
Borokers:
[FC<7 <0x80>:*>]
```

Example: Verifying Whether SSO is Configured

The following sample output shows that SSO is configured on the device:

```
Device# show redundancy states
       my state = 13 -ACTIVE
     peer state = 8  -STANDBY HOT
           Mode = Duplex
        Unit ID = 49
Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State              = sso
     Maintenance Mode = Disabled
    Manual Swact = enabled
 Communications = Up
   client count = 67
 client_notification_TMR = 30000 milliseconds
           RF debug mask = 0x0
```

Example: Displaying which SSO Protocols and Applications are Registered

The following sample output shows a list of applications and protocols that have registered as SSO protocols or applications on the device:

```
Device# show redundancy clients
 clientID = 0       clientSeq = 0        RF_INTERNAL_MSG
 clientID = 29      clientSeq = 60       Redundancy Mode RF
```

```
clientID = 139     clientSeq = 62       IfIndex
clientID = 25      clientSeq = 69       CHKPT RF
clientID = 1340    clientSeq = 90       ASR1000-RP Platform
clientID = 1501    clientSeq = 91       Cat6k CWAN HA
clientID = 78      clientSeq = 95       TSPTUN HA
clientID = 305     clientSeq = 96       Multicast ISSU Conso
clientID = 304     clientSeq = 97       IP multicast RF Clie
clientID = 22      clientSeq = 98       Network RF Client
clientID = 88      clientSeq = 99       HSRP
clientID = 114     clientSeq = 100      GLBP
clientID = 1341    clientSeq = 102      ASR1000 DPIDX
clientID = 1505    clientSeq = 103      Cat6k SPA TSM
clientID = 1344    clientSeq = 110      ASR1000-RP SBC RF
clientID = 227     clientSeq = 111      SBC RF
clientID = 71      clientSeq = 112      XDR RRP RF Client
clientID = 24      clientSeq = 113      CEF RRP RF Client
clientID = 146     clientSeq = 114      BFD RF Client
clientID = 306     clientSeq = 120      MFIB RRP RF Client
clientID = 1504    clientSeq = 128      Cat6k CWAN Interface
clientID = 75      clientSeq = 130      Tableid HA
clientID = 401     clientSeq = 131      NAT HA
clientID = 402     clientSeq = 132      TPM RF client
clientID = 5       clientSeq = 135      Config Sync RF clien
clientID = 68      clientSeq = 149      Virtual Template RF
clientID = 23      clientSeq = 152      Frame Relay
clientID = 49      clientSeq = 153      HDLC
clientID = 72      clientSeq = 154      LSD HA Proc
clientID = 113     clientSeq = 155      MFI STATIC HA Proc
clientID = 20      clientSeq = 171      IPROUTING NSF RF cli
clientID = 100     clientSeq = 173      DHCPC
clientID = 101     clientSeq = 174      DHCPD
clientID = 74      clientSeq = 183      MPLS VPN HA Client
clientID = 34      clientSeq = 185      SNMP RF Client
clientID = 52      clientSeq = 186      ATM
clientID = 69      clientSeq = 189      AAA
clientID = 118     clientSeq = 190      L2TP
clientID = 82      clientSeq = 191      CCM RF
clientID = 35      clientSeq = 192      History RF Client
clientID = 90      clientSeq = 204      RSVP HA Services
clientID = 70      clientSeq = 215      FH COMMON RF CLIENT
clientID = 54      clientSeq = 220      SNMP HA RF Client
clientID = 73      clientSeq = 221      LDP HA
clientID = 76      clientSeq = 222      IPRM
clientID = 57      clientSeq = 223      ARP
clientID = 50      clientSeq = 230      FH_RF_Event_Detector
clientID = 1342    clientSeq = 240      ASR1000 SpaFlow
clientID = 1343    clientSeq = 241      ASR1000 IF Flow
clientID = 83      clientSeq = 255      AC RF Client
clientID = 84      clientSeq = 257      AToM manager
clientID = 85      clientSeq = 258      SSM
clientID = 102     clientSeq = 273      MQC QoS
clientID = 94      clientSeq = 280      Config Verify RF cli
clientID = 135     clientSeq = 289      IKE RF Client
clientID = 136     clientSeq = 290      IPSEC RF Client
clientID = 130     clientSeq = 291      CRYPTO RSA
clientID = 148     clientSeq = 296      DHCPv6 Relay
clientID = 4000    clientSeq = 303      RF_TS_CLIENT
clientID = 4005    clientSeq = 305      ISSU Test Client
clientID = 93      clientSeq = 309      Network RF 2 Client
clientID = 205     clientSeq = 311      FEC Client
clientID = 141     clientSeq = 319      DATA DESCRIPTOR RF C
clientID = 4006    clientSeq = 322      Network Clock
clientID = 225     clientSeq = 326      VRRP
clientID = 65000   clientSeq = 336      RF_LAST_CLIENT
```

Additional References

Related Documents

Related Topic	Document Title
In-Service Software Upgrade process conceptual and configuration information	Cisco IOS XE In Service Software Upgrade Process module
Nonstop Forwarding conceptual and configuration information	Cisco Nonstop Forwarding module
Stateful switchover conceptual and configuration information	Stateful Switchover module
White paper on preforming In-Service Software Upgrades.	High-Availability Overview, Cisco IOS Software: Guide to Performing In-Service Software Upgrades
Answer to questions about the In-Service Software Upgrade product and process.	Cisco IOS In-Service Software Upgrade, Questions and Answers
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS High Availability commands	*Cisco IOS High Availability Command Reference*
Cisco IOS debug commands	*Cisco IOS Debug Command Reference*
SSO - BFD	" Bidirectional Forwarding Detection " chapter in the *IP Routing Protocols Configuration Guide*
SSO HSRP	"Configuring HSRP" chapter in the *IP Application Services Configuration Guide*
SSO - MPLS VPN 6VPE and 6PE SSO support	NSF/SSO and ISSU - MPLS VPN 6VPE and 6PE
SSO and RPR on the Cisco ASR 1000 Series Routers	*Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*
SSO VRRP	"Configuring VRRP" chapter in the *Application Services Configuration Guide*
SNMP configuration tasks	"Configuring SNMP Support" module of *Network Management Configuration Guide*
SNMP commands	*Cisco IOS Network Management Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature.	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature.	--
RFC 1907	Management Information Base for Version 2 of the Simple Network Management Protocol
RFC 2571	An Architecture for Describing SNMP Management Frameworks
RFC 2573	SNMP Applications
RFC 2574	User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)
RFC 2575	View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
RFC 2863	The Interfaces Group MIB
RFC 4133	Entity MIB (Version 3)

Technical Assistance

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

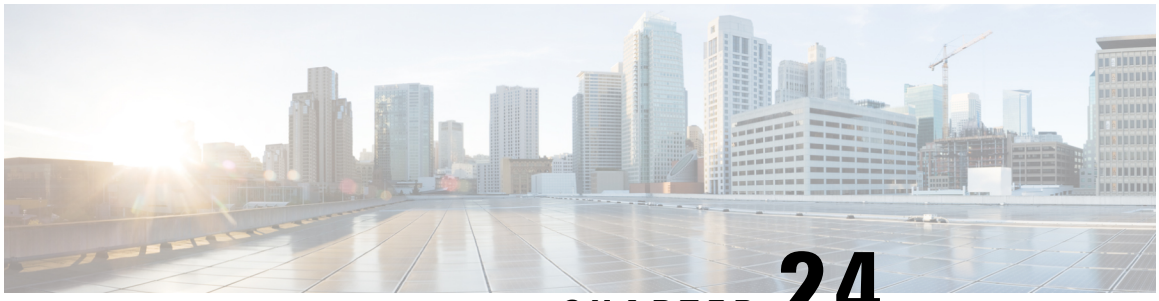
Glossary

CPE --customer premises equipment. Terminating equipment, such as terminals, telephones, and modems, supplied by the service provider, installed at customer sites, and connected to the network.

ISSU --In Service Software Upgrade. ISSU is a process that allows Cisco IOS software to be updated or otherwise modified while packet forwarding continues.

RP --Route Processor. A generic term for the centralized control unit in a chassis.

SSO --Stateful Switchover. SSO refers to the implementation of Cisco IOS software that allows applications and features to maintain a defined state between an active and standby RP. When a switching occurs, forwarding and sessions are maintained. SSO makes an RP failure undetectable to the network.



CHAPTER 24

Flexible NetFlow IPFIX Export Format

The Flexible NetFlow IPFIX Export Format feature enables sending export packets using the IPFIX export protocol. The export of extracted fields from NBAR is only supported over IPFIX.

Information About Flexible NetFlow IPFIX Export Format

Flexible NetFlow IPFIX Export Format Overview

IPFIX is an IETF standard based on NetFlow v9.

The Flexible NetFlow IPFIX Export Format feature enables sending export packets using the IPFIX export protocol. The export of extracted fields from NBAR is only supported over IPFIX.

How to Configure Flexible NetFlow IPFIX Export Format

Configuring the Flow Exporter

Perform this required task to configure the flow exporter.

Note Each flow exporter supports only one destination.

You can export to a destination using either an IPv4 or IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
6. **dscp** *dscp*
7. **source** *interface-type interface-number*
8. **output-features**
9. **template data timeout** *seconds*
10. **transport udp** *udp-port*
11. **ttl** *seconds*
12. **end**
13. **show flow exporter** *exporter-name*
14. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates the flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to the datacenter`	(Optional) Configures a description to the exporter that will appear in the configuration and the display of the **show flow exporter** command.
Step 5	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the destination system for the exporter. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**dscp** *dscp* **Example:** `Device(config-flow-exporter)# dscp 63`	(Optional) Configures differentiated services code point (DSCP) parameters for datagrams sent by the exporter. • The range for the *dscp* argument is from 0 to 63. Default: 0.

	Command or Action	Purpose
Step 7	**source** *interface-type interface-number* **Example:** `Device(config-flow-exporter)# source ethernet 0/0`	(Optional) Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 8	**output-features** **Example:** `Device(config-flow-exporter)# output-features`	(Optional) Enables sending export packets using quality of service (QoS) and encryption.
Step 9	**template data timeout** *seconds* **Example:** `Device(config-flow-exporter)# template data timeout 120`	(Optional) Configures resending of templates based on a timeout. • The range for the *seconds* argument is 1 to 86400 (86400 seconds = 24 hours).
Step 10	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 650`	Specifies the UDP port on which the destination system is listening for exported datagrams. • The range for the *udp-port* argument is from 1 to 65536.
Step 11	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 15`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. • The range for the *seconds* argument is from 1 to 255.
Step 12	**end** **Example:** `Device(config-flow-exporter)# end`	Exits flow exporter configuration mode and returns to privileged EXEC mode.
Step 13	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.
Step 14	**show running-config flow exporter** *exporter-name* **Example:** `Device# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Configuration Examples for Flexible NetFlow IPFIX Export Format

Example: Configuring Flexible NetFlow IPFIX Export Format

The following example shows how to configure IPFIX export format for Flexible NetFlow.

This sample starts in global configuration mode:

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 export-protocol ipfix
 transport udp 90
 exit
!
flow monitor FLOW-MONITOR-1
 record netflow ipv4 original-input
 exporter EXPORTER-1
!
ip cef
!
interface Ethernet 0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input
!
```

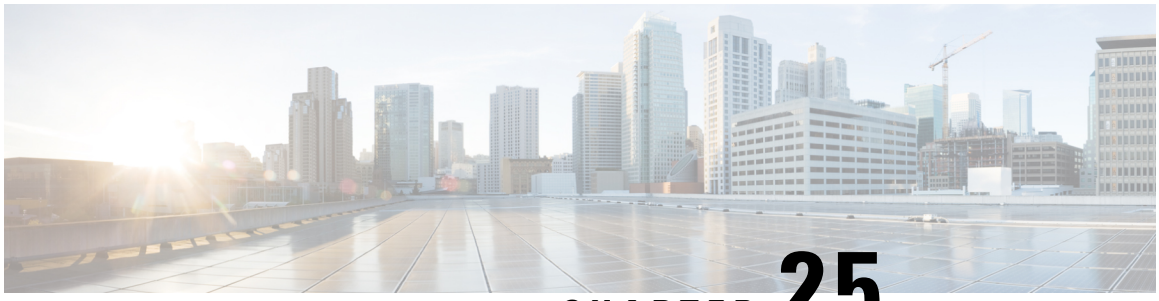
Feature Information for Flexible NetFlow: IPFIX Export Format

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 43: Feature Information for Flexible NetFlow : IPFIX Export Format

Feature Name	Releases	Feature Information
Flexible NetFlow: IPFIX Export Format	15.2(4)M Cisco IOS XE Release 3.7S 15.2(1)SY	Enables sending export packets using the IPFIX export protocol. The export of extracted fields from NBAR is only supported over IPFIX. Support for this feature was added for Cisco ASR 1000 Series Aggregation Services routers in Cisco IOS XE Release 3.7S. The following command was introduced: **export-protocol**.



CHAPTER 25

Flexible Netflow Export to an IPv6 Address

The Export to an IPv6 Address feature enables Flexible NetFlow to export data to a destination using an IPv6 address.

Information About Flexible Netflow Export to an IPv6 Address

Flexible Netflow Export to an IPv6 Address Overview

This feature enables Flexible NetFlow to export data to a destination using an IPv6 address.

How to Configure Flexible Netflow Export to an IPv6 Address

Configuring the Flow Exporter

Perform this required task to configure the flow exporter.

Note Each flow exporter supports only one destination.

You can export to a destination using either an IPv4 or IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*

5. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
6. **dscp** *dscp*
7. **source** *interface-type interface-number*
8. **output-features**
9. **template data timeout** *seconds*
10. **transport udp** *udp-port*
11. **ttl** *seconds*
12. **end**
13. **show flow exporter** *exporter-name*
14. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates the flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to the datacenter`	(Optional) Configures a description to the exporter that will appear in the configuration and the display of the **show flow exporter** command.
Step 5	**destination** {*ip-address*	*hostname*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the IP address or hostname of the destination system for the exporter. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**dscp** *dscp* **Example:** `Device(config-flow-exporter)# dscp 63`	(Optional) Configures differentiated services code point (DSCP) parameters for datagrams sent by the exporter. • The range for the *dscp* argument is from 0 to 63. Default: 0.

	Command or Action	Purpose
Step 7	**source** *interface-type interface-number* **Example:** `Device(config-flow-exporter)# source ethernet 0/0`	(Optional) Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams.
Step 8	**output-features** **Example:** `Device(config-flow-exporter)# output-features`	(Optional) Enables sending export packets using quality of service (QoS) and encryption.
Step 9	**template data timeout** *seconds* **Example:** `Device(config-flow-exporter)# template data timeout 120`	(Optional) Configures resending of templates based on a timeout. • The range for the *seconds* argument is 1 to 86400 (86400 seconds = 24 hours).
Step 10	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 650`	Specifies the UDP port on which the destination system is listening for exported datagrams. • The range for the *udp-port* argument is from 1 to 65536.
Step 11	**ttl** *seconds* **Example:** `Device(config-flow-exporter)# ttl 15`	(Optional) Configures the time-to-live (TTL) value for datagrams sent by the exporter. • The range for the *seconds* argument is from 1 to 255.
Step 12	**end** **Example:** `Device(config-flow-exporter)# end`	Exits flow exporter configuration mode and returns to privileged EXEC mode.
Step 13	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.
Step 14	**show running-config flow exporter** *exporter-name* **Example:** `Device# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Configuration Examples for Flexible Netflow Export to an IPv6 Address

Example: Configuring Multiple Export Destinations

The following example shows how to configure multiple export destinations for Flexible NetFlow for IPv4 or IPv6 traffic. This sample starts in global configuration mode:

The following example shows how to configure multiple export destinations for Flexible NetFlow for IPv4:

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 transport udp 90
 exit
!
flow exporter EXPORTER-2
 destination 172.16.10.3
 transport udp 90
 exit
!
flow record v4_r1
match ipv4 tos
match ipv4 protocol
match ipv4 source address
match ipv4 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long


flow monitor FLOW-MONITOR-1
 record v4_r1
 exporter EXPORTER-2
 exporter EXPORTER-1
!

ip cef
!
interface GigabitEthernet1/0/0
 ip address 172.16.6.2 255.255.255.0
 ip flow monitor FLOW-MONITOR-1 input

!
```

The following example shows how to configure multiple export destinations for Flexible NetFlow for IPv6:

```
!
flow exporter EXPORTER-1
 destination 172.16.10.2
 transport udp 90
 exit
!
flow exporter EXPORTER-2
 destination 172.16.10.3
 transport udp 90
```

```
 exit
!

flow record v6_r1
match ipv6 traffic-class
match ipv6 protocol
match ipv6 source address
match ipv6 destination address
match transport source-port
match transport destination-port
collect counter bytes long
collect counter packets long
!


!
flow monitor FLOW-MONITOR-2
 record v6_r1
 exporter EXPORTER-2
 exporter EXPORTER-1
!
ip cef
!
interface GigabitEthernet1/0/0
 ipv6 address 2001:DB8:2:ABCD::2/48
 ipv6 flow monitor FLOW-MONITOR-2 input
!
```

The following display output shows that the flow monitor is exporting data to the two exporters:

```
Device# show flow monitor FLOW-MONITOR-1
Flow Monitor FLOW-MONITOR-1:
  Description:       User defined
  Flow Record:       v4_r1
  Flow Exporter:     EXPORTER-1
                     EXPORTER-2
  Cache:
    Type:              normal (Platform cache)
    Status:            allocated
    Size:              4096 entries / 311316 bytes
    Inactive Timeout:  15 secs
    Active Timeout:    1800 secs
    Update Timeout:    1800 secs
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

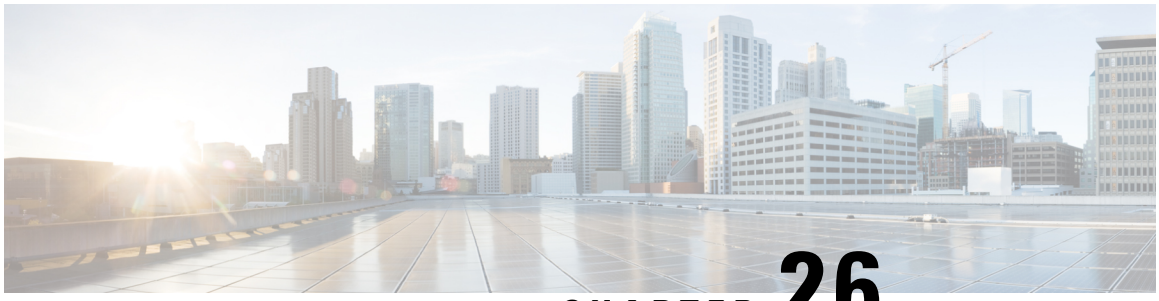
Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 26

Flexible Netflow—Egress VRF Support

The Flexible Netflow—Egress VRF Support feature enables collecting the virtual routing and forwarding (VRF) ID from outgoing packets on a router by applying an output flow monitor having a flow record that collects the VRF ID as a key or a nonkey field.

Information About Flexible Netflow Egress VRF Support

Flexible Netflow—Egress VRF Support Overview

The Flexible Netflow—Egress VRF Support feature enables collecting the virtual routing and forwarding (VRF) ID from outgoing packets on a router by applying an output flow monitor having a flow record that collects the VRF ID as a key or a nonkey field.

How to Configure Flexible Netflow Egress VRF Support

Creating a Flow Record

Perform this task to configure a customized flow record.

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one **match** criterion for use as the key field and typically has at least one **collect** criterion for use as a nonkey field.

There are hundreds of possible permutations of customized flow records. This task shows the steps that are used to create one of the possible permutations. Modify the steps in this task as appropriate to create a customized flow record for your requirements.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {**ip** | **ipv6**} {**destination** | **source**} **address**
6. Repeat Step 5 as required to configure additional key fields for the record.
7. **match flow cts** {**source** | **destination**} **group-tag**
8.
9. Repeat the above step as required to configure additional nonkey fields for the record.
10. **end**
11. **show flow record** *record-name*
12. **show running-config flow record** *record-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record FLOW-RECORD-1`	Creates a flow record and enters Flexible NetFlow flow record configuration mode. • This command also allows you to modify an existing flow record.
Step 4	**description** *description* **Example:** `Device(config-flow-record)# description Used for basic traffic analysis`	(Optional) Creates a description for the flow record.
Step 5	**match** {**ip**	**ipv6**} {**destination**	**source**} **address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	**Note** This example configures the IPv4 destination address as a key field for the record. For information about the other key fields available for the **match ipv4** command, and the other **match** commands that are available to configure key fields.
Step 6	Repeat Step 5 as required to configure additional key fields for the record.	—

	Command or Action	Purpose

Step 7	**match flow cts** {**source** | **destination**} **group-tag**
	Example:
	`Device(config-flow-record)# match flow cts source group-tag`
	`Device(config-flow-record)# match flow cts destination group-tag`
		Note This example configures the CTS source group tag and destination group tag as a key field for the record. For information about the other key fields available for the **match ipv4**/**ipv6** command, and the other **match** commands that are available to configure key fields.
		Note
		• Ingress:
		• In an incoming packet, if a header is present, SGT will reflect the same value as the header. If no value is present, it will show zero.
		• The DGT value will not depend on the ingress port SGACL configuration.
		• Egress:
		• If either propagate SGT or CTS is disabled on the egress interface, then SGT will be zero.
		• In an outgoing packet, if SGACL configuration that corresponds to the (SGT, DGT) exists, DGT will be non-zero.
		• If SGACL is disabled on the egress port/VLAN or if global SGACL enforcement is disabled, then DGT will be zero

Step 8	**Example:**	Configures the input interface as a nonkey field for the record. **Note** This example configures the input interface as a nonkey field for the record.
Step 9	Repeat the above step as required to configure additional nonkey fields for the record.	—
Step 10	**end** **Example:** `Device(config-flow-record)# end`	Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.
Step 11	**show flow record** *record-name* **Example:**	(Optional) Displays the current status of the specified flow record.

	Command or Action	Purpose
	`Device# show flow record FLOW_RECORD-1`	
Step 12	**show running-config flow record** *record-name* **Example:** `Device# show running-config flow record FLOW_RECORD-1`	(Optional) Displays the configuration of the specified flow record.

Creating a Customized Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries.

Before you begin

If you want to use a customized record instead of using one of the Flexible NetFlow predefined records, you must create the customized record before you can perform this task.

If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor. For information about the **ip flow monitor** command, refer to the *Cisco IOS Flexible NetFlow Command Reference*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]}
6. **cache** {**entries** *number* | **timeout** {**active** | **inactive** | **update**} *seconds* | **type** {**immediate** | **normal** | **permanent**}}
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **statistics packet protocol**
9. **statistics packet size**
10. **exporter** *exporter-name*
11. **end**
12. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]]
13. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**entries** *number* | **timeout** {**active** | **inactive** | **update**} *seconds* | **type** {**immediate** | **normal** | **permanent**}} **Example:** `Device(config-flow-monitor)# cache type normal`	(Optional) Modifies the flow monitor cache parameters such as timeout values, number of cache entries, and the cache type. • The values for the keywords associated with the **timeout** keyword have no effect when the cache type is set to **immediate**.
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**statistics packet protocol** **Example:** `Device(config-flow-monitor)# statistics packet protocol`	(Optional) Enables the collection of protocol distribution statistics for Flexible NetFlow monitors.
Step 9	**statistics packet size** **Example:**	(Optional) Enables the collection of size distribution statistics for Flexible NetFlow monitors.

	Command or Action	Purpose
	`Device(config-flow-monitor)# statistics packet size`	
Step 10	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 11	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 12	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status and statistics for a Flexible NetFlow flow monitor.
Step 13	**show running-config flow monitor** *monitor-name* **Example:** `Device# show running-config flow monitor FLOW_MONITOR-1`	(Optional) Displays the configuration of the specified flow monitor.

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Configuration Examples for Flexible Netflow Egress VRF Support

Example Configuring Flexible NetFlow for Egress VRF Support

The following example configures the collection of the virtual routing and forwarding (VRF) ID from outgoing packets on a router by applying an output flow monitor having a flow record that collects the VRF ID as a key field.

This example starts in global configuration mode.

```
!
flow record rm_1
match routing vrf output
match ipv4 source address
match ipv4 destination address
collect interface input
collect interface output
collect counter packets
!
flow monitor mm_1
record rm_1
!
interface GigabitEthernet 0/0/0
ip vrf forwarding green
ip address 172.16.2.2 255.255.255.252
ip flow monitor mm_1 output
!
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

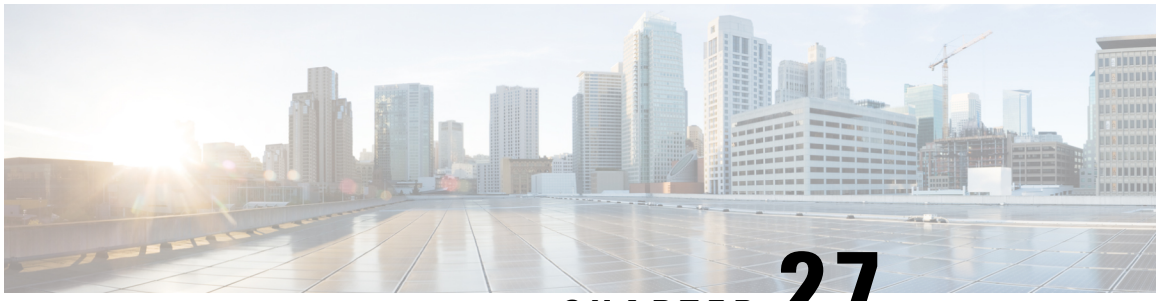
Feature Information for Flexible NetFlow—Egress VRF Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 44: Feature Information for Flexible NetFlow—Egress VRF Support

Feature Name	Releases	Feature Information
Flexible NetFlow—Egress VRF Support	Cisco IOS XE Release 3.8S	Enables collecting the virtual routing and forwarding (VRF) ID from outgoing packets on a router by applying an output flow monitor having a flow record that collects the VRF ID as a key or a nonkey field. The following commands were introduced or modified: **collect routing**, **match routing**, **option (Flexible NetFlow**, **show flow monitor**.



CHAPTER 27

Flexible NetFlow - MPLS Support

The Flexible NetFlow - MPLS Support feature supports the monitoring of the following MPLS-related fields:

- MPLS Labels 1-6 (3 bytes -- 20 bits of label, 3 bits of EXP, 1 bit of EOS).
- Top Label EXP i.e. the EXP field for label 1.
- Top Label TTL i.e. the TTL field for label 1.

Information About Flexible NetFlow MPLS Support

Flexible NetFlow—MPLS Support Overview

This feature enables collecting MPLS label IDs by applying a flow monitor having a flow record that collects the MPLS label IDs as key or nonkey fields.

How to Configure Flexible NetFlow MPLS Support

Configuring a Flow Exporter for the Flow Monitor

Perform this optional task to configure a flow exporter for the flow monitor in order to export the data that is collected by Flexible NetFlow to a remote system for further analysis and storage.

Flow exporters are used to send the data that you collect with Flexible NetFlow to a remote system such as a NetFlow Collection Engine. Exporters use UDP as the transport protocol and use the Version 9 export format.

Note Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.

You can export to a destination using either an IPv4 or IPv6 address.

Note When you configure an exporter, configure the exporter in such a way that the source interface is defined as a WAN interface. This configuration helps you prevent any unpredictable behavior because the NAT is not applied on the packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **description** *description*
5. **destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*]
6. **export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix**}
7. **transport udp** *udp-port*
8. **exit**
9. **flow monitor** *flow-monitor-name*
10. **exporter** *exporter-name*
11. **end**
12. **show flow exporter** *exporter-name*
13. **show running-config flow exporter** *exporter-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates a flow exporter and enters Flexible NetFlow flow exporter configuration mode. • This command also allows you to modify an existing flow exporter.

	Command or Action	Purpose
Step 4	**description** *description* **Example:** `Device(config-flow-exporter)# description Exports to datacenter`	(Optional) Creates a description for the flow exporter.
Step 5	**destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*] **Example:** `Device(config-flow-exporter)# destination 172.16.10.2`	Specifies the hostname or IP address of the system to which the exporter sends data. **Note** You can export to a destination using either an IPv4 or IPv6 address.
Step 6	**export-protocol** {**netflow-v5** | **netflow-v9** | **ipfix**} **Example:** `Device(config-flow-exporter)# export-protocol netflow-v9`	Specifies the version of the NetFlow export protocol used by the exporter. • Default: **netflow-v9**.
Step 7	**transport udp** *udp-port* **Example:** `Device(config-flow-exporter)# transport udp 65`	Configures UDP as the transport protocol and specifies the UDP port on which the destination system is listening for exported Flexible NetFlow traffic.
Step 8	**exit** **Example:** `Device(config-flow-exporter)# exit`	Exits Flexible NetFlow flow exporter configuration mode and returns to global configuration mode.
Step 9	**flow monitor** *flow-monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Enters Flexible NetFlow flow monitor configuration mode for the flow monitor that you created previously.
Step 10	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	Specifies the name of an exporter that you created previously.
Step 11	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 12	**show flow exporter** *exporter-name* **Example:** `Device# show flow exporter FLOW_EXPORTER-1`	(Optional) Displays the current status of the specified flow exporter.

	Command or Action	Purpose
Step 13	**show running-config flow exporter** *exporter-name* **Example:** `Device<# show running-config flow exporter FLOW_EXPORTER-1`	(Optional) Displays the configuration of the specified flow exporter.

Creating a Flow Monitor

Perform this required task to create a customized flow monitor.

Each flow monitor has a separate cache assigned to it. Each flow monitor requires a record to define the contents and layout of its cache entries. These record formats can be a user-defined format. An advanced user can create a customized format using the **flow record** command.

Before you begin

If you want to use a customized record, you must create the customized record before you can perform this task. If you want to add a flow exporter to the flow monitor for data export, you must create the exporter before you can complete this task.

Note You must use the **no ip flow monitor** command to remove a flow monitor from all of the interfaces to which you have applied it before you can modify the parameters for the **record** command on the flow monitor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name*}
6. **cache** {**timeout** {**active**} *seconds* | { **normal** }
7. Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.
8. **exporter** *exporter-name*
9. **end**
10. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]]
11. **show running-config flow monitor** *monitor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor and enters Flexible NetFlow flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.
Step 4	**description** *description* **Example:** `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	(Optional) Creates a description for the flow monitor.
Step 5	**record** {*record-name*} **Example:** `Device(config-flow-monitor)# record FLOW-RECORD-1`	Specifies the record for the flow monitor.
Step 6	**cache** {**timeout** {**active**} *seconds* | { **normal** } **Example:**	
Step 7	Repeat Step 6 as required to finish modifying the cache parameters for this flow monitor.	—
Step 8	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	(Optional) Specifies the name of an exporter that was created previously.
Step 9	**end** **Example:** `Device(config-flow-monitor)# end`	Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.
Step 10	**show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]]] **Example:** `Device# show flow monitor FLOW-MONITOR-2 cache`	(Optional) Displays the status for a Flexible NetFlow flow monitor.
Step 11	**show running-config flow monitor** *monitor-name* **Example:**	(Optional) Displays the configuration of the specified flow monitor.

	Command or Action	Purpose
	`Device# show running-config flow monitor FLOW_MONITOR-1`	

Applying a Flow Monitor to an Interface

Before it can be activated, a flow monitor must be applied to at least one interface. Perform this required task to activate a flow monitor.

While running the **ip flow monitor** command for the first interface to enable FNF monitor, you may see the following warning message displaying a GLOBAL memory allocation failure. This log is triggered by enabling FNF monitoring with a large cache size.

```
Jul  4 01:45:00.255: %CPPEXMEM-3-NOMEM: F0/0: cpp_cp_svr: QFP: 0, GLOBAL memory allocation
 of 90120448 bytes by FNF failed
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: CPR STILE
EXMEM GRAPH, Allocations: 877, Type: GLOBAL
Jul  4 01:45:00.258: %CPPEXMEM-3-TOPUSER: F0/0: cpp_cp_svr: QFP: 0, Top User: SBC, Bytes
Allocated: 53850112, Type: GLOBAL
```

The warning message does not necessarily indicate a flow monitor application failure. The warning message can indicate internal steps that FNF uses for applying memory from the EXMEM infrastructure.

To ensure that the FNF monitor is enabled successfully, use the **show flow monitor** *monitor-name* command to check **Status** (**allocated** or **not allocated**) of a flow monitor. For more information, see Displaying the Current Status of a Flow Monitor, on page 156.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. {**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**}
5. Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.
6. **end**
7. **show flow interface** *type number*
8. **show flow monitor name** *monitor-name* **cache format record**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 4	{**ip** | **ipv6**} **flow monitor** *monitor-name* {**input** | **output**} **Example:** `Device(config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.
Step 5	Repeat Steps 3 and 4 to activate a flow monitor on any other interfaces in the device over which you want to monitor traffic.	—
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 7	**show flow interface** *type number* **Example:** `Device# show flow interface GigabitEthernet 0/0/0`	Displays the status of Flexible NetFlow (enabled or disabled) on the specified interface.
Step 8	**show flow monitor name** *monitor-name* **cache format record** **Example:** `Device# show flow monitor name FLOW_MONITOR-1 cache format record`	Displays the status, statistics, and flow data in the cache for the specified flow monitor.

Configuration Examples for Flexible NetFlow MPLS Support

Example: Configuring Flexible NetFlow for MPLS Support

The following example shows how to configure a flow monitor using the Flexible NetFlow "BGP ToS next-hop" predefined record to monitor IPv4 traffic.

This sample starts in global configuration mode:

```
Router(config)#flow record mpls_1
Router(config-flow-record)#match mpls label 1 details
Router(config-flow-record)#match mpls label 1 exp
Router(config-flow-record)#match mpls label 1 ttl
```

```
Router(config-flow-record)#match mpls label 2 details
Router(config-flow-record)#match mpls label 3 details
Router(config-flow-record)#collect mpls label 4 details
Router(config-flow-record)#collect mpls label 5 details
!
flow monitor mpls_1
record mpls_1
!
interface Ethernet 0/0
 mpls flow monitor mpls_1 input
 mpls flow monitor mpls_1 output
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Flexible NetFlow conceptual information and configuration tasks	*Flexible NetFlow Configuration Guide*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Standards/RFCs

Standard	Title
No new or modified standards/RFCs are supported by this feature.	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

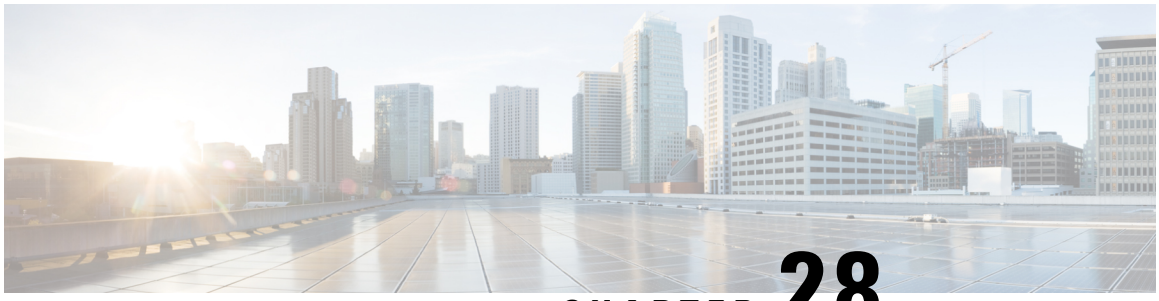
Feature Information for Flexible NetFlow: MPLS Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 45: Feature Information for Flexible NetFlow: MPLS Support

Feature Name	Releases	Feature Information
Flexible NetFlow: MPLS Support	Cisco IOS XE Release 3.9S	Enables collecting MPLS label IDs by applying a flow monitor having a flow record that collects the MPLS label IDs as key or nonkey fields. The following commands were introduced or modified: **collect mplslabel**, **match mplslabel**, **mpls flow monitor**.



CHAPTER 28

Flexible NetFlow—Prevent Export Storms

The Flexible NetFlow—Prevent Export Storms feature uses export spreading to prevent export storms that occur due to the creation of a synchronized cache. The export of the previous interval is spread during the current interval to prevent export storms.

Information About Flexible NetFlow—Prevent Export Storms

Flexible NetFlow—Prevent Export Storms Overview

The Flexible NetFlow—Prevent Export Storms feature prevents export storms at a NetFlow Collecting (NFC) device, especially when multiple Flexible NetFlow (FNF) entities are configured to export FNF records to the same NFC at the same synchronized wallclock time. Export storms occur due to the creation of the synchronized cache type. Export spreading reduces the severity of export storms and mitigates their impact.

Synchronized cache with spreading requires adding the interval timestamp field for the synchronized cache. When no spreading is configured, it is recommended to add the interval as a key, but the configuration is not rejected to maintain backward compatibility. If no export spread is specified, the default behavior is immediate export. The spread time must be smaller than half of the interval. Therefore, it will be set to half the interval time or to the configured spread interval, whichever is lower (but not lower than 1 second).

You must not enable spreading when the interval sync timeout is lower than 10 seconds (5- second spreading). This requirement comes from the need for asynchronous monitors to aggregate the data within a few seconds. Spreading might start a couple of seconds after the interval ends in order to complete the aggregation. If a synchronized interval value is lower than 10 seconds, no spreading option is visible in the command-line interface (CLI). The default spread interval, if unspecified, is 30 seconds. The maximum synchronized interval timeout value is 300 seconds. For native FNF monitors, the maximum synchronized interval timeout value could be larger. The rate calculation is provisioned as follows:

- The simple implementation is a constant rate based on the cache-size/spread-interval.
- An improved implementation is based on the current-previous-interval-cache-size/spread-interval. This provides better results when the cache is not full.

The NetFlow/IPFIX header timestamp is set to the time when the record leaves the device (and not when the record leaves the NetFlow cache). The timestamp fields in the record itself capture the timestamp of the packets and are accounted for in the NetFlow cache. A new, implementation-driven concept of a “small interval” is now implicitly introduced and understood to be directly in contrast with the concept of a “large interval”. The “large interval” can be thought of as simply the sync interval as configured by the CLI. This is the interval at the beginning of which the entire export process is to be initiated. It corresponds to the "synchronized interval" that is driven and defined by the CLI. At the beginning of a “large interval”, we must take the number of records in the cache and divide that number by the number of seconds available in which to export these records, thus yielding the calculated or derived quantity of “records per second”.

For example, if there are 100,000 records in the cache and 100 seconds in which to export these records, we would calculate and store the value 1000 records/second. Because this quantity is expressed in seconds, it follows that we will need to count the records exported in small intervals that are one second in duration.

This then, implicitly, defines the notion of a “small interval”, which is, to be succinct and equal to one second. Combining this idea of small and large intervals with the need for a state or context, it quickly becomes evident that a timer thread must be able to discern if it is beginning a “small interval” or a “large interval”.

How to Configure Flexible NetFlow—Prevent Export Storms

Configuring Flexible NetFlow—Prevent Export Storms

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor type performance-monitor** *monitor-name*
4. **cache type synchronized**
5. **cache timeout synchronized** *interval* **export-spread** *spread-interval*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor type performance-monitor** *monitor-name* **Example:** `Device(config)# flow monitor type performance-monitor my_mon`	Creates a flow monitor and enters flow monitor configuration mode. • This command also allows you to modify an existing flow monitor.

	Command or Action	Purpose
Step 4	**cache type synchronized** **Example:** `Device(config-flow-monitor)# cache type synchronized`	Configures the cache type for a Performance Monitor flow monitor.
Step 5	**cache timeout synchronized** *interval* **export-spread** *spread-interval* **Example:** `Device(config-flow-monitor)# cache timeout synchronized 12 export-spread 5`	Configures export spreading.
Step 6	**end** **Example:** `Device(config-flow-monitor)# end`	Returns to privileged EXEC mode.

Configuration Examples for Flexible NetFlow—Prevent Export Storms

Example: Flexible NetFlow—Prevent Export Storms Configuration

The following example shows how to enable and configure export spreading and prevent export storms where the synchronized interval timeout value is 12 seconds and the export spread interval is 5 seconds:

```
Device> enable
Device# configure terminal
Device(config)# flow monitor type performance-monitor my_mon
Device(config-flow-monitor)# cache type synchronized
Device(config-flow-monitor)# cache timeout synchronized 12 export-spread 5
```

Additional References for Flexible NetFlow—Prevent Export Storms

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Flexible NetFlow commands	*Cisco IOS Flexible NetFlow Command Reference*

Technical Assistance

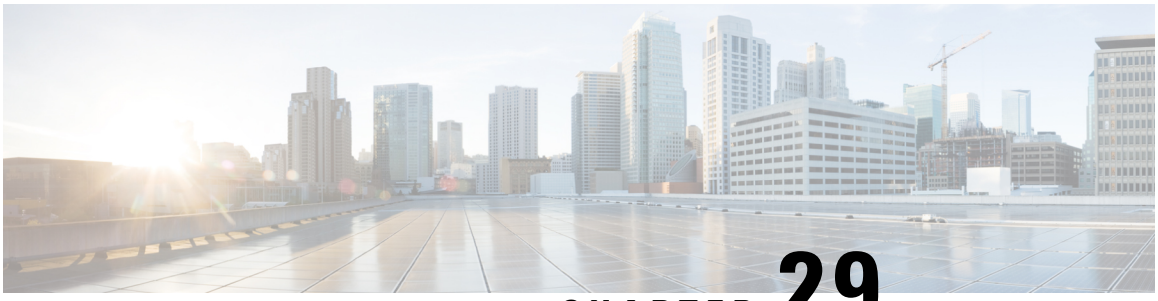
Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services,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

Feature Information for Flexible NetFlow—Prevent Export Storms

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Releases	Feature Information
Flexible NetFlow - Prevent Export Storms	Cisco IOS XE Release 3.11S	The Flexible NetFlow—Prevent Export Storms feature uses export spreading to prevent export storms that occur due to the creation of a synchronized cache. The export of the previous interval is spread during the current interval to prevent export storms.



CHAPTER 29

Flexible Packet Matching

Flexible Packet Matching (FPM) is an access control list (ACL) pattern matching tool, providing more thorough and customized packet filters. FPM enables users to match on arbitrary bits of a packet at an arbitrary depth in the packet header and payload. FPM removes constraints to specific fields that had limited packet inspection.

FPM enables users to create their own stateless packet classification criteria and to define policies with multiple actions (such as drop, log, or send Internet Control Message Protocol [ICMP] unreachable[1]) to immediately block new viruses, worms, and attacks.

Prerequisites for Flexible Packet Matching

Although access to an XML editor is not required, XML will ease the creation of protocol header description files (PHDFs).

Restrictions for Flexible Packet Matching

- FPM can search for patterns up to 32 bytes in length within the first 256 bytes of the packet.
- A maximum of 32 classes are supported in a policy-map.
- For IP option packets, FPM inspects only the fields in the Layer 2 header and the first 20 bytes of the IP header.
- For noninitial IP fragments, FPM inspects only the fields in the Layer 2 header and the first 20 bytes of the IP header.
- FPM cannot be used to mitigate an attack that requires stateful classification.

[1] Send ICMP unreachable is currently not supported on the Supervisor Engine 32 PISA.

- Because FPM is stateless, it cannot keep track of port numbers being used by protocols that dynamically negotiate ports. Thus, when using FPM, port numbers must be explicitly specified.
- FPM cannot perform IP fragmentation or TCP flow reassembly.
- FPM inspects only IPv4 unicast packets.
- FPM cannot classify packets with IP options.
- FPM does not support multicast packet inspection.
- FPM is not supported on tunnel and MPLS interfaces.
- Noninitial fragments will not be matched by the FPM engine.
- Offset can be only a constant in a match start construct.
- FPM cannot match across packets.
- Mapping of FPM policies to control-plane is not supported.

Information About Flexible Packet Matching

Flexible Packet Matching Functional Overview

FPM allows customers to create their own filtering policies that can immediately detect and block new viruses and attacks.

A filtering policy is defined via the following tasks:

- Load a PHDF (for protocol header field matching)
- Define a class map and define the protocol stack chain (traffic class)
- Define a service policy (traffic policy)
- Apply the service policy to an interface

Protocol Header Description File

Protocol headers are defined in separate files called PHDFs; the field names that are defined within the PHDFs are used for defining the packet filters. A PHDF is a file that allows the user to leverage the flexibility of XML to describe almost any protocol header. The important components of the PHDF are the version, the XML file schema location, and the protocol field definitions. The protocol field definitions name the appropriate field in the protocol header, allow for a comment describing the field, provide the location of the protocol header field in the header (the offset is relative to the start of the protocol header), and provide the length of the field. Users can choose to specify the measurement in bytes or in bits.

Note The total length of the header must be specified at the end of each PHDF.

Note When redundant sup PHDF files are used by FPM policy, the files should also be on standby sup's corresponding disk. If the files are not available FPM policy will not work after the switch over.

Users can write their own custom PHDFs via XML for existing or proprietary protocols. However, the following standard PHDFs can also be loaded onto the router via the **load protocol** command: ip.phdf, ether.phdf, tcp.phdf, and udp.phdf.

Note Because PHDFs are defined via XML, they are not shown in a running configuration. However, you can use the **show protocol phdf** command to verify the loaded PHDF.

Standard PHDFs are available on Cisco.com at the following URL:
http://www.cisco.com/cgi-bin/tablebuild.pl/fpm

Filter Description

A filter description is a definition of a traffic class that can contain the header fields defined in a PHDF (using the **match field** command). If a PHDF is not loaded, the traffic class can be defined through the datagram header start (Layer 2) or the network header start (Layer 3) (using the **match start** command). If a PHDF has been loaded onto the router, the class specification begins with a list of the protocol headers in the packet.

A filter definition also includes the policy map; that is, after a class map has been defined, a policy map is needed to bind the match to an action. A policy map is an ordered set of classes and associated actions, such as drop, log, or send ICMP unreachable.

For information on how to configure a class map and a policy map for FPM, see the How to Configure a Flexible Packet Matching Traffic Class and Traffic Policy section.

How to Configure Flexible Packet Matching

Creating a Traffic Class for Flexible Packet Matching

Note If the PHDF protocol fields are referenced in the access-control classmap, the stack classmap is required in order to make FPM work properly

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **load protocol** *location:filename*
4. **class-map** [**type** {**stack** | **access-control**}] *class-map-name* [**match-all** | **match-any**]
5. **description** *character-string*

6. **match field** *protocol protocol-field* {**eq** [*mask*] | **neq** | [*mask*] | **gt** | **lt** | **range** *range* | **regex** *string*} *value* [**next** *next-protocol*]
7. **match start** {**l2-start** | **l3-start**} **offset** *number* **size** *number* {**eq** | **neq** | **gt** | **lt** | **range** *range* | **regex** *string*} {*value* [*value2*] | [*string*]}
8. **match class** *class-name* [**packet-range** *low high* | **byte-range** *low high*] **session**
9. **exit**
10. **exit**
11. **show class-map** [**type** {**stack** | **access-control**} | *class-map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**load protocol** *location:filename* **Example:** `Router(config)# load protocol disk2:udp.phdf`	(Optional) Loads a PHDF onto a router. • The specified location must be local to the router. **Note** If a PHDF is not loaded, only the **match start** command can be used; that is, you cannot issue the **match field** command. **Note** For the ASR platform, PHDF files should be manually copied (through the **load protocol** command) to the active and standby route processor (RP) file systems.
Step 4	**class-map** [**type** {**stack**	**access-control**}] *class-map-name* [**match-all**	**match-any**] **Example:** `Router(config)# class-map type access-control c1`	Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. • **type stack** -- Enables FPM to determine the correct protocol stack in which to examine. • **type access-control** -- Determines the exact pattern to look for in the protocol stack of interest. • *class-map-name* -- Can be a maximum of 40 alphanumeric characters. • If **match-all** or **match-any**or are not specified, traffic must match all the match criterion to be classified as part of the traffic class.

	Command or Action	Purpose
Step 5	**description** *character-string* **Example:** `Router(config-cmap)# description "match on slammer packets"`	(Optional) Adds a description to the class map.
Step 6	**match field** *protocol protocol-field* {**eq** [*mask*] | **neq** | [*mask*] | **gt** | **lt** | **range** *range* | **regex** *string*} *value* [**next** *next-protocol*] **Example:** `Router(config-cmap)# match field udp dest-port eq 0x59A`	(Optional) Configures the match criteria for a class map on the basis of the fields defined in the PHDFs. • The **next** *next-protocol* keyword-argument pair is available only after configuring the **class-map type stack** command.
Step 7	**match start** {**l2-start** | **l3-start**} **offset** *number* **size** *number* {**eq** | **neq** | **gt** | **lt** | **range** *range* | **regex** *string*} {*value* [*value2*] | [*string*]} **Example:** `Router(config-cmap)# match start l3-start offset 224 size 4 eq 0x4011010`	(Optional) Configures the match criteria for a class map on the basis of the datagram header (Layer 2) or the network header (Layer 3).
Step 8	**match class** *class-name* [**packet-range** *low high* | **byte-range** *low high*] **session** **Example:** `Router(config-cmap)# match class c2 packet-range 1 5 session`	(Optional) Configures match criteria for a class map that identifies a session (flow) containing packets of interest, which is then applied to all packets transmitted during the session. The **packet-range** and **byte-range** keywords create a filter mechanism that increases the performance and matching accuracy of regex-based FPM class maps by classifying traffic that resides in the narrow packet number or packet byte ranges of each packet flow. When the **session** keyword is used with the *class-name* argument, the classification results are preserved for the subsequent packets of the same packet session. When the **session** keyword is used with the **packet-range** or **byte-range** keywords, the classification results are preserved for the specified packets or bytes of the same packet session.
Step 9	**exit** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.
Step 10	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Router(config)# exit`	
Step 11	**show class-map** [**type** {**stack** | **access-control**} | *class-map-name*] **Example:** `Router# show class-map type access-control slammer`	(Optional) Displays configured FPM class maps.

Troubleshooting Tips

To track all FPM events, issue the **debug fpm event** command.

The following sample output is from the **debug fpm event**command:

```
*Jun 21 09:22:21.607: policy-classification-inline(): matches class: class-default *Jun 21
 09:22:21.607: packet-access-control(): policy-map: fpm-policy, dir: input, match. retval:
 0x0, ip-flags: 0x80000000
```

What to Do Next

After you have defined at least one class map for your network, you must create a traffic policy and apply that policy to an interface as shown in the following task “Creating a Traffic Policy for Flexible Packet Matching.”

Creating a Traffic Policy for Flexible Packet Matching

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map type access-control** *policy-map-name*
4. **description** *character-string*
5. **class** *class-name* **insert-before** *class-name*
6. **drop** [**all**]
7. **log** [**all**]
8. **service-policy** *policy-map-name*
9. **exit**
10. **interface** *type number*
11. **service-policy type access-control** {**input** | **output**} *policy-map-name*
12. **exit**
13. **exit**
14. **show policy-map** [**type access-control** | **interface** *type number* | **input** | **output**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map type access-control** *policy-map-name* **Example:** `Router(config)# policy-map type access-control fpm-udp-policy`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters policy-map configuration mode.
Step 4	**description** *character-string* **Example:** `Router(config-pmap)# description "policy for UDP based attacks"`	(Optional) Adds a description to the policy map.
Step 5	**class** *class-name* **insert-before** *class-name* **Example:** `Router(config-pmap)# class slammer`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command. The **class** command also classifies traffic to the traffic policy and enters policy-map class configuration mode. • The **insert-before** *class-name* keyword and argument adds a class map to any location within the policy map. If this option is not issued, the class map is appended to the end of the policy map.
Step 6	**drop** [**all**] **Example:** `Router(config-pmap-c)# drop all`	(Optional) Configures a traffic class to discard packets belonging to a specific class. The **all** keyword is used to discard the entire stream of packets belonging to the traffic class. If this command is issued, note the following restrictions: • Discarding packets is the only action that can be configured in a traffic class. • When a traffic class is configured with the **drop** command, a "child" (nested) policy cannot be configured for this specific traffic class through the **service policy** command.

	Command or Action	Purpose
		• Discarding packets cannot be configured for the default class specified via the **class class-default** command. • If the **drop all**command is specified, then this command can only be associated with a **class map type access-control** command.
Step 7	**log** [**all**] **Example:** `Router(config-pmap-c)# log all`	(Optional) Generates log messages for the traffic class. The **all** keyword is used to log the entire stream of discarded packets belonging to the traffic class. This keyword is only available for a class map that is created with the **class-map type access-control** command.
Step 8	**service-policy** *policy-map-name* **Example:** `Router(config-pmap-c)# service policy fpm-udp-policy`	Creates hierarchical service policies.

Step 9	**exit**
	Example:
	`Router(config-pmap-c)# exit`
	Example:
	`Router(config-pmap)# exit`
		Exits policy-map class configuration mode and policy-map configuration mode.

Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitEthernet 0/1`	Configures an interface type and enters interface configuration mode.
Step 11	**service-policy type access-control** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy type access-control input fpm-policy`	Specifies the type and the name of the traffic policy to be attached to the input or output direction of an interface.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 13	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`Router(config)# exit`	
Step 14	**show policy-map** [**type access-control** | **interface** *type number* | **input** | **output**] **Example:** `Router# show policy-map type access-control interface gigabitethernet 0/1`	(Optional) Verifies the FPM configuration. **Note** Once a traffic policy is created for FPM, a matched packet can be copied or redirected to a different destination interface.

Configuration Examples for an FPM Configuration

Configuring and Verifying FPM on ASR Platform: Example

The following example shows how to configure FPM on the ASR platform.

```
load protocol bootflash:ip.phdf
load protocol bootflash:tcp.phdf
class-map type stack match-all ip_tcp
 match field IP protocol eq 6 next TCP
class-map type access-control match-all test_class
 match field TCP dest-port gt 10
 match start l3-start offset 40 size 32 regex "ABCD"
policy-map type access-control child
 class test_class
  drop
policy-map type access-control parent
 class ip_tcp
  service-policy child
interface GigabitEthernet0/3/0
 ip address 10.1.1.1 255.0.0.0
 service-policy type access-control input parent
```

In the following sample output, all TCP packets are seen under the class-map “ip_tcp” and all packets matching the specific pattern are seen under the class-map “test_class.” TCP packets without the specific pattern are seen under the child policy “class-default,” while all non-TCP packets are seen under the parent policy “class-default.” (The counter is 0 in this example.)

```
Router# show policy-map type access-control interface GigabitEthernet0/3/0
GigabitEthernet0/3/0
 Service-policy access-control input: parent
  Class-map: ip_tcp (match-all)
  2024995578 packets, 170099628552 bytes
  5 minute offered rate 775915000 bps
  Match: field IP version eq 4
  Match: field IP ihl eq 5
  Match: field IP protocol eq 6 next TCP
 Service-policy access-control : child
 Class-map: test_class (match-all)
  1598134279 packets, 134243279436 bytes
  5 minute offered rate 771012000 bps, drop rate 771012000 bps
  Match: field TCP dest-port gt 10
  Match: start l3-start offset 40 size 32 regex "ABCD"
```

```
 drop
 Class-map: class-default (match-any)
  426861294 packets, 35856348696 bytes
  5 minute offered rate 4846000 bps, drop rate 0 bps
  Match: any
 Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
Router#
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	*Cisco IOS Security Command Reference*
Configuring FPM using traffic classification definition files.	"Flexible Packet Matching XML Configuration" module in the *Cisco IOS Security Configuration Guide: Securing the Data Plane*
Complete suite of quality of service (QoS) commands	*Cisco IOS Quality of Service Solutions Command Reference*

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

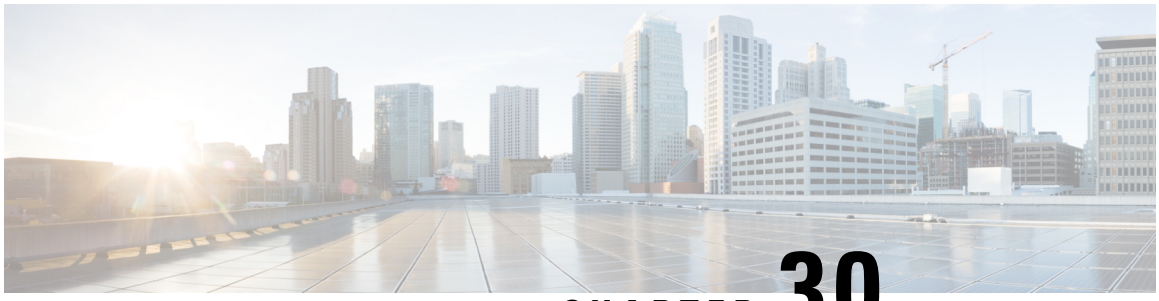
Feature Information for Flexible Packet Matching

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 46: Feature Information for Flexible Packet Matching

Feature Name	Releases	Feature Information
Flexible Packet Matching	Cisco IOS XE Release 2.2	FPM is a packet classification feature that allows users to define one or more classes of network traffic by pairing a set of standard matching operators with user-defined protocol header fields. The following commands were introduced or modified: **class (policy-map)** **class-map** **debug fpm event, description (class-map)** **load protocol** **match field** **match start**, **policy-map**, **service-policy**, **show class-map**, **show policy-map interface**, **show protocol phdf**.



CHAPTER 30

Cisco Data Collection Manager

The Cisco Data Collection Manager (DCM) feature provides a scalable data collection framework for collecting data from Cisco devices. The DCM supports a profile-based configuration that allows users to set parameters for collecting, processing, and exporting the data. The DCM also supports a flexible data process mechanism that allows users to derive information from raw data, and calculate baselines, summaries, statistical distribution, and percentiles. The integrated monitoring feature of DCM allows users to monitor multiple data sources, such as Simple Network Management Protocol (SNMP) MIB objects and **show** command output. The DCM supports the export of collected data based on transfer configurations through multiple data export mechanisms, such as FTP, TFTP, and Internet Protocol Flow Information Export (IPFIX).

Information About Cisco Data Collection Manager

Cisco Data Collection Manager

Overview of Cisco Data Collection Manager

Cisco Data Collection Manager (DCM) is an efficient and reliable data collection agent that is embedded in managed devices, such as routers and switches. DCM works on a push model, which is based on a subscribe-and-notify data pattern, as opposed to the pull model, which is based on a request-and-response data pattern, in traditional SNMP-based network management.

Configuration and Deployment

The Bulkstat client application is implemented using the DCM core services to retrieve data and export it to the user. The Bulkstat client provides the only available user interface for DCM access. The client also provides CLI access through a new set of configuration commands and MIB access through the CISCO-DATA-COLLECTION-MIB. You can configure the data to be exported as a file. Also, you can configure the data to be processed and the processed file to be collected by the NMS.

Data Collection

The Data Collection Manager (DCM) provides data subscription services for multiple data sources, such as the Simple Network Management Protocol (SNMP) MIB objects and the output of **show** commands. The DCM allows you to configure the data that needs to be collected. The DCM also allows you to specify when and at what interval the data should be collected.

Data Processing

The Data Collection Manager (DCM) supports advanced on-board data processing that includes baseline calculation, summary calculation, statistical distribution, and percentile computation. The DCM is integrated with CISCO-EXPRESSION-MIB to externally create and customize MIB objects for monitoring and to support the CLI to define expressions.

Data Export and Retrieval

The Data Collection Manager (DCM) provides data retrieval management to ensure that the data collection does not impact device resources. The DCM can export data in a file format using multiple export protocols such as FTP, TFTP, Secure copy protocol (SCP), and Secure File Transfer Protocol (SFTP). The DCM provides a query mechanism with which data can be selectively exported based on the configured time interval and other selection criteria. The DCM application also provides data filtering services and exports the filtered data. You can also set primary and secondary destinations for exporting the collected data in a raw or processed format. Snapshots of the collected data can be stored for later retrieval.

Performance Management Solutions

The Data Collection Manager (DCM) can be used to manage various aspects of performance management. The following list provides a few scenarios.

Traffic Engineering

The primary goal of traffic engineering is to control traffic flow and provide Class of Service (CoS) and Quality of Service (QoS) to the end user, while using network resources optimally. It is the responsibility of the network management server (NMS) to provide low-level management through traffic conditioning, packet shaping, queue management, and other functions that regulate traffic flow through the network. The DCM can be used to collect data with a high granularity to help the NMS make dynamic traffic engineering decisions.

Capacity Planning and Trend Analysis

The key priorities for network operators today are acceleration in new service introduction, reduction in the complexity of deployment and management of services, and reduction in the capital or operational overhead expenditure. These services require unique capabilities that are specific to a particular network operator and the ability to provide specific service characteristics on a per-consumer basis. It is important to identify trends in the network traffic, forecast future traffic levels, assess whether the existing network capacity is sufficient to handle the projected load, and automatically redesign the network to support future traffic demands. The DCM can be used to collect resource variables that are important for effective capacity trend information, such as memory, queue depth, broadcast volume, buffer, Frame Relay congestion notification, and backplane utilization.

Diagnostics

The streaming function of the DCM can be used for real-time troubleshooting. Also, the DCM facilitates data retrieval for selective periods and aids in troubleshooting.

SLA Management

A service level agreement (SLA) includes a what-if analysis for network changes and application changes, baselining and trending for defined performance variables, exception management for defined capacity and performance variables, and QoS management. The DCM can be used to collect periodic data for reporting purposes.

Bulkstat

Two challenges that network providers usually face are data gathering and data analysis. Network providers need to gather large volumes of data to analyze the performance of the network and to have operational control over their network. Large service providers are strengthening their data gathering and analysis infrastructure. Traditionally, Simple Network Management Protocol (SNMP) agents are used to expose management data on managed systems. But, SNMP is not well suited for gathering large volumes of data, especially over short time intervals.

For example, service providers charge customers depending on the network usage. Also this data must be available on customer request. Accounting applications based on SNMP polling models consume significant network bandwidth because they poll large volumes of data frequently. The SNMP protocol data unit (PDU) is a complex data type specific to SNMP and is expensive to process because the SNMP objects and tables must be sorted in a lexicographic order. All the entries in SNMP MIB tables are lexicographically ordered by their object identifiers, because there is an implied ordering in the MIB based on the order of the object identifiers.

In such cases, the need to continuously poll large or bulk SNMP statistics can be avoided by using applications known as collectors to retrieve data.

The Bulkstat application is one such collector that uses the services of the Data Collection Manager (DCM) to provide the following functions:

- Collecting SNMP MIB object values and the output of **show** commands.
- Processing the collected data to create summary, percentiles, and auto-baselined values.
- Exporting collected data through simple file transfer.
- Scheduling calendar events for data collection and export.

The Bulkstat application provides command-line access through a set of new configuration commands and exclusive MIB access through CISCO-DATA-COLLECTION-MIB to collect SNMP data.

You can configure Bulkstat for the following functions:

- Specify the way Bulkstat retrieves bulk statistics.
- Specify the time interval in seconds at which Bulkstat transfers data to receivers.
- Specify the maximum size of the bulk statistics file.
- Specify the context, instance, and period at which the system retrieves bulk statistics.
- Configure file-related parameters.
- Configure the interface type on which you want to collect statistics.
- View the parameters that Bulkstat uses to collect statistics by using the **show bulkstat** commands.

Bulkstat Configuration Elements

The following list shows the elements that you can configure using the Bulkstat interface:

Data Set

This section describes the data set elements that you can configure to collect Simple Network Management Protocol (SNMP) data and CLI data. Only objects having the same index elements can be grouped in a single object list.

SNMP Data

The SNMP data set contains the following fields:

Table 47: SNMP Data Elements

Name	Description	Configuration Status
Objects	Specifies the object to be collected. Multiple objects can be configured to form a data set. The textual name of the object can be used for configuring an object. If the device does not recognize the textual name, the object identifier (OID) format can be used for configuring the name.	Mandatory
Object alias	Specifies the optional alias name that each object can have.	Optional

Note Only objects having the same index elements can be grouped in a single data set. For example, the objects *ifDescr* and *cieIfSpeedReceive* belong to different tables, but they are indexed by *ifIndex*. Hence the two objects can be grouped in the same data set. *ifDescr* and *entPhysicalDescr* cannot be grouped in the same data set, because *ifDescr* is indexed by *ifIndex* and *entPhysicalDescr* is indexed by *entPhysicalIndex*.

Objects from tables having sparse dependency can also be grouped in the same data set. Similar SNMP data elements can be linked to multiple subscriptions. Each subscription can collect different entries for the objects based on the instance configuration and context configuration.

CLI Data

The CLI data set contains the following fields:

Table 48: CLI Data Elements

Name	Description	Configuration Status	
CLI	Specifies the CLI command for which the **show** output needs to be collected. More than one CLI can be specified in the same data set.	Mandatory	

Instance Set

This section specifies the instance set elements that you can configure to collect Simple Network Management Protocol (SNMP) data. More than one instance of the same type can be added to the set. Combinations of types of instance set elements are not supported.

The SNMP Instance set contains the following fields:

Table 49: SNMP Instance Elements

Name	Description	Configuration Status
Exact	Specifies the instance for which the data should be collected. More than one instance can be specified, but only fully qualified instances should be specified.	Optional
Wildcard	Specifies all instances for all objects under the object configured in the data set.	Optional
Range	Specifies the start and end instances. All instances within the range, including the start and end, are collected, but only fully qualified instances should be specified.	Optional
Repetition	Specifies the start of the repetition and the number of repetitions. All instances from the start until the number of repetitions within the subtree are collected.	Optional
Interface	Specifies the interface instead of the index. The *ifIndex* assigned to the interface will be used as an index. This can be used for MIB objects indexed by *ifIndex*.	Optional

Filter Set

This section describes the filter configuration per object.

The filter set elements that you can configure to collect Simple Network Management Protocol (SNMP) data are described here. More than one filter of the same type can be added to the set.

Table 50: Filter Elements

Name	Description	Status
Object match	Specifies the value to be used to match against the value retrieved for the object during collection. The value provided needs to match the type of the object. If there is an error in the type matching, the configuration is not accepted. More than one value can be specified for an object, and more than one object can have matching values.	Optional

Process Set

For detailed information, see the topic “Data Processing.”

Data Group

This section describes the data group, which contains the data-group name, data-group type, data set, instance set, filter set, polling interval, SNMP context, and other processing options.

Table 51: Data Group Elements

Name	Description	Configuration Status
Data	Specifies any one of the data types as defined in the topic "Data Set."	Mandatory
Instance	Specifies any one of the instance types as defined in the topic "Instance Set."	Optional, if not specified. Default behavior of the instance set is wildcard. Only applicable for SNMP.
Filter	Specifies any one of the filter types as defined in the topic "Filter Set."	Optional, if not specified. Only applicable for SNMP.
Polling interval	Specifies the collection periodic interval in seconds. In case of recurring collection, the data is collected at the expiration of the collection interval until the collection is stopped.	Optional Default value is 600 seconds.
Context	Specifies the management context from which to obtain data for this data group.	Optional
Process summary	Enables summary processing of the data marked to be processed in the corresponding data-set configuration.	Optional
Process distribution	Enables distribution processing of the data marked to be processed in the corresponding data-set configuration.	Optional
Process percentile	Enables percentile processing of the data marked to be processed in the corresponding data-set configuration.	Optional
Process auto-baseline	Enables auto-baselining processing of the data marked to be processed in the corresponding data-set configuration. If auto-baseline process is enabled, the other processes, such as summary, distribution, and percentile configurations, if done previously, are removed because auto-baseline process uses these functionalities internally. **Note** Removing this configuration will not reinstate the other configurations that are removed.	Optional

Name	Description	Configuration Status
Discard raw	Specifies whether to store raw data. If data is processed, the user can choose to store only process data by setting the option.	Optional

Data Profile

This section describes the data profile that is used to group multiple data groups. This is done to simplify the configuration and to aggregate data of similar nature. A data profile can have multiple data groups. A data group can have constraints in the data specified in the element. If two sets of data need to be written to the same file, the respective data groups should be linked as part of a single profile.

The data profile has the following fields:

Table 52: Data Profile Elements

Name	Description	Status
Data groups	Specifies the data group to be linked to this profile. Multiple data groups can be linked to a single profile.	Mandatory before activating a profile
Transfer interval	Specifies the transfer periodic interval in seconds. In case of recurring transfer, the data is transferred when the transfer interval expires.	Optional Default interval transfer raw data is 1800.
Process interval	Specifies the process periodic interval in seconds. The data is processed during every collection interval as soon as it is collected. When the process interval expires, the processed data is written into a file and transferred.	Optional Default interval transfer processes data is 3600.
Primary URL	Specifies the URL of the primary management station. The files containing the collected data are transferred to this URL when the transfer interval expires.	Mandatory
Secondary URL	Specifies the URL of the secondary management station to be used in case the transfer to the primary management station fails.	Optional
Schema	Specifies the file data format. The schemaASCII option is supported.	Optional Default format is schemaAscii.

Name	Description	Status
Retry	Specifies the number of times that the transfer is retried in case of transfer failures to both primary and secondary management stations. This command has an effect only if the **retain** command is configured in the profile. The retry interval is computed by dividing the retention time by the number of retries. For example, if the file is retained for 60 minutes and the retry is 6 times, the transfer is attempted every 10 minutes, until the transfer succeeds or the file is removed.	Optional Default retry value is retry is 3.
Buffer-size	Specifies the maximum size to which the file containing the collected data can grow. When it reaches the limit, the file is closed and the transfer is attempted based on the transfer configuration associated with the data group or profile.	Optional
Retention memory	Specifies the time, in seconds, to retain the file in the memory.	Optional
Retention USB	Specifies the time, in seconds, to retain the file in the USB. This option is available only if the device supports the USB drive.	Optional

Resource Limit

The Bulkstat application allows you to configure memory resource limit in percentage. Bulkstat will deactivate all profiles if the remaining memory is less than specified limit.

By default, the resource limit is set at 95 percentage of the total available memory to accommodate for high memory usage on certain platforms.

Calendar Scheduling

The Bulkstat application allows you to schedule each subscription for collection. A subscription can be scheduled for one-time collection or periodic collection. A periodic subscription can be repeated infinitely or for a specified number of repetitions. A timer is instantiated for every activated subscription.

Table 53: Calendar Scheduling Elements

Name	Description	Configuration Status
One shot	Specifies that the data is collected for a specified collection interval.	Optional
Recurring	Specifies that the data is collected regularly at the specified time, day, month, and for a specified collection interval.	Optional

Predefined Data Sets and Data Groups

In NG3K devices, a Simple Network Management Protocol (SNMP) MIB table having large number of objects will result in large number of lines of bulkstat configuration needed to collect these objects. For example, if the SNMP table contains 40 objects, then we need 41 lines of configuration for bulkstat data set, 2 lines for an instance set and a minimum of 3 lines for a data group. So, there would be a total of 46 lines of configuration

required to collect a SNMP table containing 40 objects. This can slow down the system as it takes a longer time to generate the output of Cisco IOS nonvolatile generation commands and results in a large output.

Configuring predefined data-sets and data-groups as required by the users is a solution for this issue as predefined configurations do not appear in the output of **show bulkstat** commands. The predefined data-sets, data-groups, and instance-sets would be present only if all the SNMP objects in the set are present in the SNMP table.

Note

- All predefined configuration names start with a prefix _pd_.
- You cannot create data-sets, data-groups, and instance-sets with the same prefix.
- The predefined configurations cannot be modified using CLI interface or using MIB.

Following is an example of predefined data-sets and data-groups:

- bulkstat data _pd_MobileStationDS type snmp
- bulkstat data _pd_MobileStationStatsDS type snmp
- bulkstat data _pd_ClientDS type snmp
- bulkstat data _pd_MobilityDS type snmp
- bulkstat data-group _pd_MobileStationDG
- bulkstat data-group _pd_MobileStationStatsDG
- bulkstat data-group _pd_ClientDG
- bulkstat data-group _pd_MobilityDG

Displaying Predefined Configurations

The following three commands are used to display the predefined data-sets and data-groups:

- **show bulkstat data-group**—Shows bulkstat data-group details.
- **show bulkstat pre-defined** —Shows all predefined configurations.
- **show bulkstat profile**—Shows bulkstat profile details.

SNMP Data Collection

The Data Collection Manager (DCM) is used to collect the Simple Network Management Protocol (SNMP) MIB Object data. This collection is supported through both Bulkstat CLI and CISCO-DATA-COLLECTION-MIB. This allows a network management server (NMS) application to configure a set of MIB objects and a set of instances whose values need to be collected on a periodic basis. You can configure the collected data to be periodically sent to the NMS through file export.

CLI Data Collection

You can configure the Data Collection Manager (DCM) module to collect **show** command output through the Bulkstat CLI. The DCM captures the raw **show** command output and periodically exports the data through the ASCII file format.

Data Processing

Data processing allows users to derive information from raw SNMP data, by calculating summaries and percentiles.

Service providers rely on monitored SNMP data to alert network management systems (NMSs) of changing network conditions. By periodically monitoring the device data and comparing it against a set of thresholds, the network can automatically alert the operators, thereby allowing efficient operations.

Summary

You can enable summary processing on the collected object value and calculate minimum, maximum, and average values. A summary is calculated for only those objects that are marked as process capable in the data group and uses the absolute or delta value as per the object configuration.

Distribution

You can enable distribution processing on the collected object value by specifying the object type, minimum value, maximum value, and the number of buckets to distribute the value. Based on the configuration, counters are maintained per bucket and are incremented whenever the data falls into a bucket range.

Percentile

You can enable percentile processing on the collected object value. A percentile is calculated on every process interval expiry. Distribution configuration is mandatory to enable percentile processing. Percentile computation is done assuming that the distribution is normal.

Auto-baseline

You can enable baseline processing on the collected object value. The baseline internally uses all summary, distribution, and percentile calculations to provide baseline values. You can configure either baseline processing or other forms of processing, such as summary, distribution, and percentile calculations.

The auto-baseline feature in DCM calculates the baseline values for variables of interest on the device and allows network management applications or network operators to retrieve the baseline values. The baseline values can be displayed in terms of percentiles or a median with standard deviation.

File Data Export

The file data export feature on the Data Collection Manager (DCM) exports the collected data based on the transfer configurations. Data can be exported in various formats, and Bulkstat files are one such format to collect data. The format in which the data is inserted into the file conforms to the schemaAscii format described in CISCO-DATA-COLLECTION-MIB and CISCO-BULK-FILE-MIB. The data sequence in which the data is stored is determined based on the sequence in which the data is received.

The Cisco File Transfer module is responsible for transferring the files as per the transfer configuration. This module interfaces with the Cisco IOS IFS module to transfer the file to the specified URL. A file can be retained in the device whether the transfer was a success or a failure.

File names are created using the following format:

- Raw data file name: <profile-name>_<host-name>_raw_<time when the file is created in "%y%m%d_%H%M%S%k">
- Processed data file name: <profile-name>_<host-name>_proc_<time when the file is created in "%y%m%d_%H%M%S%k">

How to Configure Cisco Data Collection Manager

Configuring an SNMP Bulkstat Data Set

The first step in configuring the Simple Network Management Protocol (SNMP) periodic data collection and transfer mechanism is to configure one or more data sets. A data set is used to group objects of similar types, based on the data source. The data set is defined outside of the data group. This external definition gives the user the flexibility to use the same data set across multiple data groups and to collect the output for different instances and different contexts.

Note All objects in an SNMP data set must be indexed by the same MIB index. However, the objects in the data set must not belong to the same MIB or the MIB table.

Perform this task to configure the SNMP Bulkstat data set.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bulkstat data** *data-set-name* **type snmp**
4. **object** *oid* [**alias** *alias-name*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**bulkstat data** *data-set-name* **type snmp** **Example:** `Device(config)# bulkstat data interface-stats type snmp`	Defines an SNMP Bulkstat data set and enters SNMP bulk statistics data set configuration mode. The creation of an SNMP Bulkstat data set creates a row in the *cdcDGBaseObjectEntry* table in the SNMP MIB. **Note** As predefined data sets begin with an underscore, you cannot configure data sets starting with an underscore.

Step 4	**object** *oid* [**alias** *alias-name*]
	Example:
	`Device(config-bs-ds-snmp)# object 1.3.6.1.2.1.2.2.1.11`
	Example:
	`Device(config-bs-ds-snmp)# object ifDescr`
	Example:
	`Device(config-bs-ds-snmp)# object ifInOctets`
	Example:
	`Device(config-bs-ds-snmp)# object 1.3.6.1.2.1.2.2.1.10 alias ifInOctets`
	Example:
	`Device(config-bs-ds-snmp)# object 1.3.6.1.2.1.2.2.1.10 alias interfaceInBytes`
		Adds a MIB object to the SNMP Bulkstat data set. If the object is already present in the data set, this command replaces the old object configuration with the new configuration.
		• Repeat this command until all objects to be monitored are added to this list.

Step 5	**end** **Example:** `Device(config-bs-ds-snmp)# end`	Exits SNMP Bulkstat data set configuration mode and returns to privileged EXEC mode.

Configuring an SNMP BulkStat Instance Set

The Simple Network Management Protocol (SNMP) instance set specifies the instances for which the data should be collected. Each subscription can collect different entries for specified objects based on the instance configuration. While more than one instance of the same type can be added to the instance set, a combination of different types is not supported.

Perform this task to configure the SNMP Bulkstat instance set.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bulkstat instance** *instance-set-name* **type snmp**
4. **exact oid** *oid*
5. **exact interface** *interface-id*
6. **wildcard**
7. **wildcard oid** *oid*
8. **wildcard interface** *interface-id*
9. **repetition oid** *oid* **max** *value*
10. **range start** *oid* **end** *oid*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bulkstat instance** *instance-set-name* **type snmp** **Example:** `Device(config)# bulkstat instance exact type snmp`	Defines an SNMP Bulkstat instance set and enters SNMP Bulkstat instance set configuration mode. The creation of an SNMP Bulkstat instance set creates a row in the *cdcDGInstanceEntry* table in the SNMP MIB. **Note** An instance created using this command can be linked to more than one data group. **Note** As predefined instance sets begin with an underscore, you cannot configure instance sets starting with an underscore.

Step 4	**exact oid** *oid*
	Example:
	`Device(config-bs-is-snmp)# exact oid 1.2.3`
	Example:
	`Device(config-bs-is-snmp)# exact oid 1`
		(Optional) Indicates that the specified instance, when appended to the object list, is the complete OID.

	Command or Action	Purpose
Step 5	**exact interface** *interface-id* **Example:** `Device(config-bs-is-snmp)# exact interface Gigabitethernet 0/0 sub-if`	(Optional) Specifies an interface name and number, for example interface Ethernet 0, instead of specifying the ifIndex OID for the interface.
Step 6	**wildcard** **Example:** `Device(config-bs-is-snmp)# wildcard`	(Optional) Specifies whether an object used for evaluating an expression is to be wildcarded during an event configuration.

Step 7	**wildcard oid** *oid*
	Example:
	`Device(config-bs-is-snmp)# wildcard oid 1.2.3`
	Example:
	`Device(config-bs-is-snmp)# wildcard oid 1`
		(Optional) Indicates that all subindices of the specified OID belong to this schema.

Step 8	**wildcard interface** *interface-id* **Example:** `Device(config-bs-is-snmp)# wildcard interface Gigabitethernet 0/0 sub-if`	(Optional) Specifies an interface name and number, for example interface Ethernet 0, instead of specifying the ifIndex OID for the interface.
Step 9	**repetition oid** *oid* **max** *value* **Example:** `Device(config-bs-is-snmp)# repetition oid 1.2.3.4 max 2000`	(Optional) Configures data collection to repeat *get-next* for the maximum number of instances starting from the specified *oid* instance.
Step 10	**range start** *oid* **end** *oid* **Example:** `Device(config-bs-is-snmp)# range start 1.2.3.4 end 1.2.3.6`	(Optional) Configures a range of instances for which the data is collected.
Step 11	**end** **Example:** `Device(config-bs-is-snmp)# end`	Exits SNMP Bulkstat instance set configuration mode and returns to privileged EXEC mode.

Configuring an SNMP BulkStat Filter Set

The Simple Network Management Protocol (SNMP) filter set specifies the filter configuration for every SNMP object.

Perform this task to configure the SNMP Bulkstat filter set.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **bulkstat filter** *filter-set-name*
4. **match** *object-name* {**eq** *line* | **start** *line* | **not** {**eq** *line* | **start** *line*}}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bulkstat filter** *filter-set-name* **Example:** `Device(config)# bulkstat filter ifType`	Defines an SNMP Bulkstat filter set and enters SNMP Bulkstat filter set configuration mode.

Step 4	**match** *object-name* {**eq** *line* | **start** *line* | **not** {**eq** *line* | **start** *line*}}
	Example:
	`Device(config-bs-fs)# match ifType eq 6767`
	Example:
	`Device(config-bs-fs)# match ifDescr start "Ethernet"`
	Example:
	`Device(config-bs-fs)# match ifType not eq 2`
		(Optional) Specifies a value to be used to match against the value retrieved for the object during collection.
		• More than one value can be specified for an object, and more than one object can have match values.
		• Enclose strings in quotes.

Step 5	**end** **Example:** `Device(config-bs-fs)# end`	Exits SNMP Bulkstat filter set configuration mode and returns to privileged EXEC mode.

Configuring a Command BulkStat Data Set

The command Bulkstat data set specifies the **show** commands for which the output is to be collected. You can specify more than one command in the same data set.

Perform this task to add **show** commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bulkstat data** *data-set-name* **type command**
4. **add cmd** *command-line*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bulkstat data** *data-set-name* **type command** **Example:** `Device(config)# bulkstat data show-snmp type command`	Defines a command Bulkstat data set and enters command Bulkstat data set configuration mode. **Note** Data-sets of any type cannot be created with _pd_ prefix.

Step 4	**add cmd** *command-line*
	Example:
	`Device(config-bs-ds-cmd)# add cmd show snmp`
	Example:
	`Device(config-bs-ds-cmd)# add cmd show ip interface brief`
		Adds a **show** command for which the output needs to be collected.
		• Add as many **show** commands as needed to this list.

Step 5	**end** **Example:** `Device(config-bs-ds-cmd)# end`	Exits command Bulkstat data set configuration mode and returns to privileged EXEC mode.

Configuring a BulkStat Data Group

The Bulkstat data group element is used to group the data set, filter set, and instance set and also to specify the processing options.

Perform this task to configure the Bulkstat data group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bulkstat data-group** *data-group-name*
4. **collect type** {**command** | **snmp**} {{**data** *data-set-name* **filter** *filter-set-name*} | **instance** *instance-set-name*}
5. **context** *context-name*
6. **interval polling** *polling-interval*
7. **discard**
8. **process**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bulkstat data-group** *data-group-name* **Example:** `Device(config)# bulkstat data-group if-dg`	Defines a Bulkstat data group and enters Bulkstat data group configuration mode. • The creation of a Simple Network Management Protocol (SNMP) Bulkstat data group creates a row in the *cdcDGEntry* table in the SNMP MIB. **Note** Data-groups of any type cannot be created with _pd_ prefix.
Step 4	**collect type** {**command**	**snmp**} {{**data** *data-set-name* **filter** *filter-set-name*}	**instance** *instance-set-name*} **Example:** The following example shows how to configure the Bulkstat to collect the output of the **show snmp** command with filter set. `Device(config-bs-dg)# collect type command data show-snmp filter Ethernet` **Example:** The following example shows how to configure the Bulkstat to collect SNMP objects.	

	Command or Action	Purpose

	`Device(config-bs-dg)# collect type snmp data interface-stats instance ins-exact filter ifType`
	Example:
	The following example shows how to configure the Bulkstat to collect SNMP objects.
	`Device(config-bs-dg)# collect type snmp data User-Stats`
	Note When the instance set is not specified, the predefined default instance set, _pd_wildcardIS, will be used.	

Step 5	**context** *context-name* **Example:** `Device(config-bs-dg)# context ctx-name`	Specifies the management context from which to obtain data for this data group.
Step 6	**interval polling** *polling-interval* **Example:** `Device(config-bs-dg)# interval polling 100`	Specifies the collection periodic interval in seconds. In case of recurring collection, the data is collected at the expiration of the collection interval until the collection is stopped.
Step 7	**discard** **Example:** `Device(config-bs-dg)# discard`	Specifies whether to discard the raw data.
Step 8	**process** **Example:** `Device(config-bs-dg)# process`	Configures process-related parameters for a data group.
Step 9	**end** **Example:** `Device(config-bs-dg)# end`	Exits Bulkstat data group configuration mode and returns to privileged EXEC mode.

Configuring a Bulkstat Profile

The profile element is used to group multiple data groups. This grouping simplifies the configuration and aggregates data of a similar nature. If two sets of data need to be written to the same file, the respective data groups should be linked as part of a single profile.

Perform this task to configure the Bulkstat profile:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bulkstat profile** *profile-name*
4. **data-group** { *data-group-name* | *pre-defined-data-group-name* [**interval polling** {*seconds*}]}
5. **interval transfer** {**process** | **raw** }{*seconds*}}
6. **file format schemaASCII**
7. **file retain** {**disk** *url* | **memory** *seconds*}
8. **file size** *bytes*
9. **file transfer** {**retry** *number* | **url** {**primary** *url* | **secondary** *url* }}
10. **enable**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device > enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bulkstat profile** *profile-name* **Example:** `Device(config)# bulkstat profile if-stats`	Creates a profile with the given name and enters Bulkstat profile configuration mode. If the profile is already created, this command sets the context for the existing profile.
Step 4	**data-group** { *data-group-name* | *pre-defined-data-group-name* [**interval polling** {*seconds*}]} **Example:** `Device(config-bs-profile)# data-group if-dg`	Specifies the data group to be linked to this profile. Multiple data groups can be linked to a single profile. Interval polling of pre-defined data-groups can be configured at the profile level, and this will modify interval polling of the actual pre-defined data-group.

Step 5	**interval transfer** {**process** | **raw** }{*seconds*}}
	Example:
	`Device(config-bs-profile)# interval transfer raw 100`
	Example:
	`Device(config-bs-profile)# interval transfer process 4000`
		Specifies the transfer periodic interval in seconds. In case of recurring transfer, the data is transferred at the expiration of the transfer interval until the transfer is stopped.
		• The **process** keyword specifies the process periodic interval in seconds. The data is processed during every collection interval as soon as it is collected. When the process interval expires, the processed data is written to a file and transferred.

	Command or Action	Purpose
Step 6	**file format schemaASCII** **Example:** `Device(config-bs-profile)# file format schemaASCII`	Configures the file-related parameter for a profile. • Specifies the file data format in schemaASCII.
Step 7	**file retain** {**disk** *url* | **memory** *seconds*} **Example:** `Device(config-bs-profile)# file retain memory 2000`	Configures the file-related parameter **retain**for a profile. • **disk**—Retains the file in the specified location in the disk for a specified amount of time in seconds. • **memory**—Retains the file in the memory for a specified amount of time in seconds.
Step 8	**file size** *bytes* **Example:** `Device(config-bs-profile)# file size 2048`	Configures the file-related parameter for a profile. • **size**—Specifies the maximum buffer size in bytes. When the limit is reached, the file is closed and transfer is attempted based on the transfer configuration associated with the data group or the profile.
Step 9	**file transfer** {**retry** *number* | **url** {**primary** *url* | **secondary** *url* }} **Example:** `Device(config-bs-profile)# file transfer url primary tftp://10.0.0.1/dcm/cpu-stats`	Configures the file related parameter **transfer** for a profile. • **primary**—Specifies the URL of the primary management station. The files containing the collected data are transferred to this URL when the transfer interval expires. • **secondary**—Specifies the URL to be used in case the transfer to the primary management station fails.
Step 10	**enable** **Example:** `Device(config-bs-profile)# enable`	Enables the profile for collection and transfer.
Step 11	**end** **Example:** `Device(config-bs-profile)# end`	Exits Bulkstat data profile configuration mode and returns to privileged EXEC mode.

Configuring Bulkstat Calendar Scheduling

Perform this task to configure Bulkstat calendar scheduling.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **bulkstat schedule** *schedule-name* **at** *time-of-day* {*date* | *week* | *month date* {**oneshot** | **recurring**} | **oneshot** | **recurring**}
4. **profile** *profile-name* **start** {**oneshot** | **recurring** *number*}
5. **profile** *profile-name* **stop**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device > enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bulkstat schedule** *schedule-name* **at** *time-of-day* {*date*	*week*	*month date* {**oneshot**	**recurring**}	**oneshot**	**recurring**} **Example:** `Device(config)# bulkstat schedule mycal at 18:30 feb 05 oneshot`	Defines the Bulkstat calendar scheduler set and enters Bulkstat event scheduler configuration mode. **Note** Choose one of the options to configure the event scheduler. You can configure these options only in global configuration mode.
Step 4	**profile** *profile-name* **start** {**oneshot**	**recurring** *number*} **Example:** `R1(config-bs-schedule)# profile cpu-process start recurring 5`	Creates a profile and sets the condition to enable the profile for a one-time event or enables the profile for multiple events.
Step 5	**profile** *profile-name* **stop** **Example:** `R1(config-bs-schedule)# profile cpu-process stop`	Disables the profile.
Step 6	**end** **Example:** `Device(config-bs-schedule)# end`	Exits Bulkstat event scheduler configuration mode and returns to privileged EXEC mode.

Configuring a Bulkstat Resource Limit

Perform this task to configure memory resource limit in percentage of the total available memory.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bulkstat resource limit memory** *memory-usage-percentage*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bulkstat resource limit memory** *memory-usage-percentage* **Example:** `Device(config)# bulkstat resource limit memory 45`	Defines the memory resource limit. **Note** To allow for high system memory usage on some platforms, the default resource limit is set to 95 percentage of the total available memory. The range is 20 to 100 percentage.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for Cisco Data Collection Manager

Example: Collecting Sorted CPU Processes

The following is sample output that shows the configuration tasks that you need to perform to collect data about the processes in the cpu in the device, such as a router or a switch in a sorted order. The polling interval is set for 300 seconds, while the interval at which the raw data is transferred is set at 1500 seconds.

```
Device> enable
Device# configure terminal
Device(config)# bulkstat data show-cpu type command
Device(config-bs-ds-cmd)# add cmd show processes cpu sorted
Device(config-bs-ds-cmd)# exit
Device(config)# bulkstat data-group show-cpu
```

```
Device(config-bs-dg)# collect type command data show-cpu
Device(config-bs-dg)# interval polling 300
Device(config-bs-dg)# process
Device(config-bs-dg-proc)# exit
Device(config-bs-dg)# exit
Device(config)# bulkstat profile show-cpu
Device(config-bs-profile)# data-group show-cpu
Device(config-bs-profile)# interval transfer raw 1500
Device(config-bs-profile)# enable
Device(config-bs-profile)# end
Device#
```

Example: Collecting SNMP Interface Statistics

The following shows the configuration steps that you need to perform to collect SNMP interface statistics for the specified interface, *Ethernet*. The polling interval is set at 30 seconds, while the interval at which the raw data is transferred is set at 60 seconds.

```
Device> enable
Device# configure terminal
Device(config)# bulkstat data if-mib type snmp
Device(config-bs-ds-snmp)# object ifDescr
Device(config-bs-ds-snmp)# object ifInOctets
Device(config-bs-ds-snmp)# object ifOutOctets
Device(config-bs-ds-snmp)# exit
Device(config)# bulkstat instance if-mib type snmp
Device(config-bs-is-snmp)# wildcard
Device(config-bs-ds-snmp)# exit
Device(config)# bulkstat filter if-mib
Device(config-bs-fs)# match ifDescr start "Ethernet"
Device(config-bs-fs)# exit
Device(config)# bulkstat data-group if-group
Device(config-bs-dg)# interval polling 30
Device(config-bs-dg)# collect type snmp data if-mib instance if-mib filter if-mib
Device(config-bs-dg)# exit
Device(config)# bulkstat profile snmp_profile
Device(config-bs-profile)# file transfer url primary tftp://10.64.68.12/dcm_data/
Device(config-bs-profile)# interval transfer raw 60
Device(config-bs-profile)# data-group if-group
Device(config-bs-profile)# enable
Device(config-bs-profile)# end
Device#
```

Example: Configuring the Processing Show Commands Output

The following shows the configuration steps that you need to perform and to display the result of the **show data-group** command:

```
Device> enable
Device# configure terminal
Device(config)# bulkstat data ds1 type snmp
Device(config-bs-ds-snmp)# object 1.3.6.1.2.1.2.2.1.16 alias ifOutOctets
Device(config-bs-ds-snmp)# object 1.3.6.1.2.1.2.2.1.10 alias ifInOctets
Device(config-bs-ds-snmp)# exit
Device(config)# bulkstat instance is1 type snmp
```

```
Device(config-bs-is-snmp)# wildcard
Device(config-bs-ds-snmp)# exit
Device(config)# bulkstat data-group dg1
Device(config-bs-dg)# collect type snmp data ds1 instance is1
Device(config-bs-dg)# interval polling 15
Device(config-bs-dg)# process
Device(config-bs-dg-proc)# enable percentile
Device(config-bs-dg-proc)# object 1.3.6.1.2.1.2.2.1.16 sample absolute
Device(config-bs-dg-proc-obj)# buckets 250
Device(config-bs-dg-proc-obj)# exit
Device(config-bs-dg-proc)# exit
Device(config-bs-dg)# exit
Device(config)# bulkstat profile profile1
Device(config-bs-profile)# data-group dg1
Device(config-bs-profile)# interval transfer process 1800
Device(config-bs-profile)# file size 1024000
Device(config-bs-profile)# enable
Device(config-bs-profile)# end
Device# show bulkstat data-group dg1 process summary

Data-Group dg1 Process data

Objectname       : 1.3.6.1.2.1.2.2.1.16

Number of samples: 4

Summary Data     :
Instance         Min     Max     Sum         Average
-----------------------------------------------------------
436752384           11376     11376     45504     11376.000
436867072           11376     11376     45504     11376.000
436871168           11376     11376     45504     11376.000
436887552           11376     11376     45504     11376.000
436809728           11376     11376     45504     11376.000
436830208           11376     11376     45504     11376.000
436920320           11376     11376     45504     11376.000
436891648           11376     11376     45504     11376.000
436793344           11376     11376     45504     11376.000
436899840           11376     11376     45504     11376.000
436817920           11376     11376     45504     11376.000
436768768           11376     11376     45504     11376.000
436785152           11376     11376     45504     11376.000
436772864           11376     11376     45504     11376.000
436736000           11376     11376     45504     11376.000
436731904           11376     11376     45504     11376.000
436838400           11376     11376     45504     11376.000
436858880           11376     11376     45504     11376.000
436756480           11376     11376     45504     11376.000
436740096           11376     11376     45504     11376.000
436776960           11376     11376     45504     11376.000
436834304           11376     11376     45504     11376.000
436875264           11376     11376     45504     11376.000
436924416           11376     11376     45504     11376.000

Device# show bulkstat data-group dg1 process distribution

Data-Group dg1 Process data

Objectname       : 1.3.6.1.2.1.2.2.1.16

Number of samples: 9

Distribution Data:
* Buckets with no data are not shown
```

```
Instance      Number of   Range           Range
              buckets     start            end
-------------------------------------------------------
436752384       250         0           4294967295

Bucket    Bucket          Bucket         Count
Index     Start           End
-------------------------------------------------------
1         0.000        17179869.180        9

Instance     Number of     Range          Range
             buckets       start           end
-------------------------------------------------------
436867072       250         0          4294967295

Bucket    Bucket         Bucket         Count
Index     Start          End
-------------------------------------------------------
1         0.000        17179869.180        9

Device# show bulkstat data-group dg1 process percentile

Data-Group dg1 Process data

Objectname       : 1.3.6.1.2.1.2.2.1.16

Number of samples: 11


Percentile Data:
Instance         : 436752384
Percentile Values:
P[000.000] = 11376.000
P[000.130] = 11376.000
P[002.280] = 11376.000
P[015.870] = 11376.000
P[050.000] = 11376.000
P[084.130] = 11376.000
P[097.720] = 11376.000
P[099.870] = 11376.000
P[100.000] = 11376.000

Instance         : 436867072
Percentile Values:
P[000.000] = 11376.000
P[000.130] = 11376.000
P[002.280] = 11376.000
P[015.870] = 11376.000
P[050.000] = 11376.000
P[084.130] = 11376.000
P[097.720] = 11376.000
P[099.870] = 11376.000
P[100.000] = 11376.000

Instance         : 436871168
Percentile Values:
P[000.000] = 11376.000
P[000.130] = 11376.000
P[002.280] = 11376.000
P[015.870] = 11376.000
P[050.000] = 11376.000
P[084.130] = 11376.000
P[097.720] = 11376.000
```

```
P[099.870] = 11376.000
P[100.000] = 11376.000
```

Additional References for Cisco Data Collection Manager

Related Documents

Related Topic	Document Title
BSDCM commands: Complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS BSDCM Command Reference
Periodic MIB data collection and transfer mechanism	*SNMP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
IETF Standard	*Exporting MIB Variables using the IPFIX Protocol*
RFC 2982	*Distributed Management Expression MIB*

MIBs

MIB	MIBs Link
• CISCO-BULK-FILE-MIB • CISCO-DATA-COLLECTION-MIB • CISCO-PROCESS-MIB • Expression-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco Data Collection Manager

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 54: Feature Information for Cisco Data Collection Manager

Feature Name	Releases	Feature Information
Cisco Data Collection Manager		DCM 2.0 provides users an option to collect management data from various sources (SNMP, CLI), optionally process the data (max, min, avg, and percentile), and export the collected/processed/summarized data in multiple export formats (ASCII file, IPFIX stream, and so on). DCM 2.0 replaces the traditional PULL (polling) model with a PUSH model for collection of management information.

Glossary

DATA GROUP—A collection of data set, instance set, filter set, and process set.

DATA SET—A collection of data-related configurations. A data set can be of multiple types: SNMP, command, and expression.

FILTER SET—A collection of filter-related configurations.

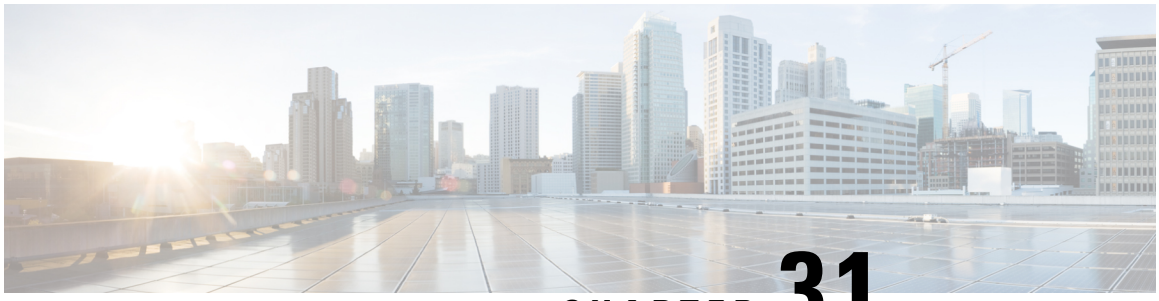
INSTANCE SET—A collection of instance-related configurations for SNMP.

PROCESS SET—A collection of data-processing-related configurations.

PROFILE—A collection of data group, transfer, and storage.

STORAGE—A collection of storage-related configurations.

TRANSFER—A collection of transfer-related configurations.



CHAPTER 31

Telnet Access over IPv6

The Telnet client and server in the Cisco software support IPv6 connections.

Prerequisites for Telnet Access over IPv6

To enable Telnet access over IPv6 to a device, you must create a vty interface and password.

Information About Telnet Access over IPv6

Telnet Access over IPv6

The Telnet client and server in Cisco software support IPv6 connections. A user can establish a Telnet session directly to the device using an IPv6 Telnet client, or an IPv6 Telnet connection can be initiated from the device. A vty interface and password must be created in order to enable Telnet access to an IPv6 device.

How to Enable Telnet Access over IPv6

Enabling Telnet Access to an IPv6 Device and Establishing a Telnet Session

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 host** *name* [*port*] *ipv6-address*

4. **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
5. **password** *password*
6. **login** [**local** | **tacacs**]
7. **ipv6 access-class** *ipv6-access-list-name* {**in** | **out**]
8. **telnet** *host* [*port*] [*keyword*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 host** *name* [*port*] *ipv6-address* **Example:** `Device(config)# ipv6 host cisco-sj 2001:DB8:20:1::12`	Defines a static hostname-to-address mapping in the hostname cache.
Step 4	**line** [**aux**	**console**	**tty**	**vty**] *line-number* [*ending-line-number*] **Example:** `Device(config)# line vty 0 4`	Creates a vty interface.
Step 5	**password** *password* **Example:** `Device(config)# password hostword`	Creates a password that enables Telnet.
Step 6	**login** [**local**	**tacacs**] **Example:** `Device(config)# login tacacs`	(Optional) Enables password checking at login.
Step 7	**ipv6 access-class** *ipv6-access-list-name* {**in**	**out**] **Example:** `Device(config)# ipv6 access-list hostlist`	(Optional) Adds an IPv6 access list to the line interface. • Using this command restricts remote access to sessions that match the access list.
Step 8	**telnet** *host* [*port*] [*keyword*] **Example:**	Establishes a Telnet session from a device to a remote host using either the hostname or the IPv6 address.

	Command or Action	Purpose
	`Device(config)# telnet cisco-sj`	• The Telnet session can be established to a device name or to an IPv6 address.

Configuration Examples for Telnet Access over IPv6

Examples: Enabling Telnet Access to an IPv6 Device

The following examples provide information on how to enable Telnet and start a session to or from an IPv6 device. In the following example, the IPv6 address is specified as 2001:DB8:20:1::12, and the hostname is specified as cisco-sj. The **show host** command is used to verify this information.

```
Device# configure terminal
Device(config)# ipv6 host cisco-sj 2001:DB8:20:1::12
Device(config)# end
Device# show host
Default domain is not set
Name/address lookup uses static mappings
Codes:UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined
Host                 Port  Flags      Age Type   Address(es)
cisco-sj             None  (perm, OK)  0  IPv6 2001:DB8:20:1::12
```

To enable Telnet access to a device, create a vty interface and password:

```
Device(config)# line vty 0 4
password lab
login
```

To use Telnet to access the device, you must enter the password:

```
Device# telnet cisco-sj
Trying cisco-sj (2001:DB8:20:1::12)... Open
User Access Verification
Password:
cisco-sj
.
.
.
verification
```

It is not necessary to use the **telnet** command. Specifying either the hostname or the address is sufficient, as shown in the following examples:

```
Device# cisco-sj
```

or

```
Device# 2001:DB8:20:1::12
```

To display the IPv6 connected user (line 130) on the device to which you are connected, use the **show users** command:

```
Device# show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
 130 vty 0                 idle                 00:00:22   8800::3
```

Note that the address displayed is the IPv6 address of the source of the connection. If the hostname of the source is known (either through a domain name server [DNS] or locally in the host cache), then it is displayed instead:

```
Device# show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
 130 vty 0                 idle                 00:02:47   cisco-sj
```

If the user at the connecting device suspends the session with ^6x and then enters the **show sessions** command, the IPv6 connection is displayed:

```
Device# show sessions
Conn Host                Address             Byte  Idle Conn Name
*  1 cisco-sj 2001:DB8:20:1::12      0     0 cisco-sj
```

The Conn Name field shows the hostname of the destination only if it is known. If it is not known, the output might look similar to the following:

```
Device# show sessions
Conn Host                Address             Byte  Idle Conn Name
*  1 2001:DB8:20:1::12 2001:DB8:20:1::12      0     0 2001:DB8:20:1::12
```

Additional References for IPv6 Source Guard and Prefix Guard

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
IPv4 addressing	*IP Addressing: IPv4 Addressing Configuration Guide*
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

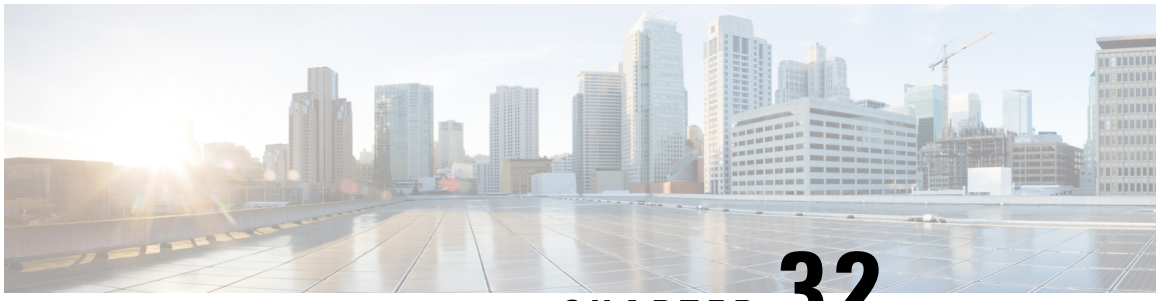
Feature Information for Telnet Access over IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 55: Feature Information for Telnet Access over IPv6

Feature Name	Releases	Feature Information
Telnet Access over IPv6	12.2(2)T 12.2(18)SXE 12.2(25)SEA 12.2(25)SG 12.2(33)SRA 15.0(2)SG Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.2SG	Telnet access over IPv6 is supported. The following commands were introduced or modified: **ipv6 access-class**, **ipv6 host**.



CHAPTER 32

IPv6 Support for TFTP

TFTP uses UDP over IPv4 or IPv6 as its transport and can work over IPv4 and IPv6 network layers.

Information About IPv6 Support for TFTP

TFTP IPv6 Support

TFTP is designed to transfer files over the network from one host to another using the most minimal set of functionality possible. TFTP uses a client/server model in which clients can request to copy files to or from a server. TFTP uses UDP over IPv4 or IPv6 as its transport, and it can work over IPv4 and IPv6 network layers.

TFTP File Downloading for IPv6

IPv6 supports TFTP file downloading and uploading using the **copy** command. The **copy** command accepts a destination IPv6 address or IPv6 hostname as an argument and saves the running configuration of the device to an IPv6 TFTP server, as follows:

```
Device# copy running-config tftp://[3ffe:xxxx:c18:1:290:27ff:fe3a:9e9a]/running-config
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

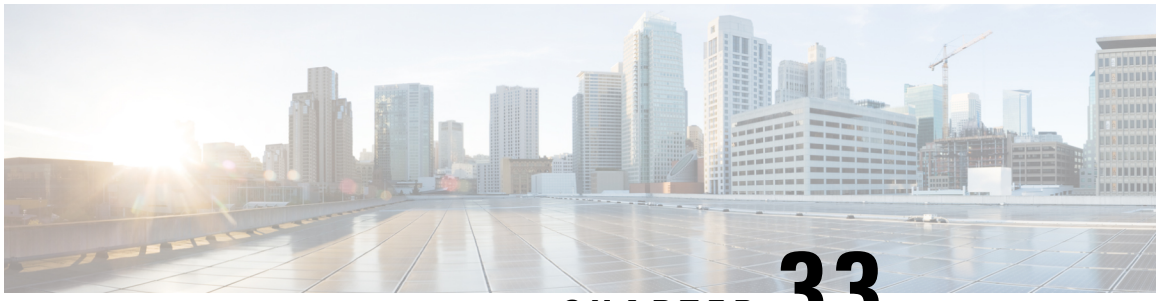
Feature Information for IPv6 Support for TFTP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 56: Feature Information for IPv6 Support for TFTP

Feature Name	Releases	Feature Information
TFTP IPv6 Support		IPv6 support for TFTP is supported. No commands were introduced or modified.



CHAPTER 33

SSH Support Over IPv6

Secure Shell (SSH) provides support for IPv6 addresses that enable a Cisco device to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

Prerequisites for SSH Support over IPv6

- An IPsec (Data Encryption Standard [DES] or 3DES) encryption software image is loaded on your device. IPv6 transport for the SSH server and SSH client requires an IPsec encryption software image.
- A hostname and host domain are configured for your device.
- A Rivest, Shamir, and Adelman (RSA) key pair, which automatically enables SSH, is generated for your device.
- A user authentication mechanism for local or remote access is configured on your device.
- To authenticate SSH clients, configure TACACS+ or RADIUS over an IPv4 transport and then connect to an SSH server over an IPv6 transport.

The basic restrictions for SSH over an IPv4 transport apply to SSH over an IPv6 transport. The use of locally stored usernames and passwords is the only user authentication mechanism supported by SSH over an IPv6 transport. TACACS+ and RADIUS user authentication mechanisms are not supported over an IPv6 transport.

Information About SSH Support over IPv6

SSH over an IPv6 Transport

Secure shell (SSH) SSH in IPv6 functions the same and offers the same benefits as SSH in IPv4. The SSH server feature enables an SSH client to make a secure, encrypted connection to a Cisco device, and the SSH

client feature enables a Cisco device to make a secure, encrypted connection to another Cisco device or to any other device running an SSH server. IPv6 enhancements to SSH consist of support for IPv6 addresses that enable a Cisco device to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

How to Enable SSH Support over IPv6

Enabling SSH on an IPv6 Device

This task is optional. If you do not configure SSH parameters, then the default values will be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh** [**timeout** *seconds* | **authentication-retries** *integer*]
4. **exit**
5. **ssh** [**-v** { **1** | **2** } | **c** { **3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc** } | **-l** *userid* | **-l** *userid:vrfname number ip-address ip-address* | **-l** *userid:rotary number ip-address* | **-m** { **hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96** } | **-o numberofpasswordprompts** *n* | **-p** *port-num*] { **ip-addr** | **hostname**} [**command** | **-vrf**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip ssh** [**timeout** *seconds*	**authentication-retries** *integer*] **Example:** `Device(config)# IP ssh timeout 100 authentication-retries 2`	Configures SSH control variables on your device.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits configuration mode, and returns the device to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**ssh** [**-v** { **1** | **2** } | **c** { **3des** | **aes128-cbc** | **aes192-cbc** | **aes256-cbc** } | **-l** *userid* | **-l** *userid:vrfname number ip-address ip-address* | **-l** *userid:rotary number ip-address* | **-m** { **hmac-md5** | **hmac-md5-96** | **hmac-sha1** | **hmac-sha1-96** } | **-o** **numberofpasswordprompts** *n* | **-p** *port-num*] { **ip-addr** | **hostname**} [**command** | **-vrf**] **Example:** `Device# ssh -l userid1 2001:db8:2222:1044::72`	Starts an encrypted session with a remote networking device.

Configuration Examples for SSH Support over IPv6

Example: Enabling SSH on an IPv6 Device

```
Device# configure terminal
Device(config)# ip ssh
Device(config)# exit
Device(config)# ssh -l userid1 2001:db8:2222:1044::72
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

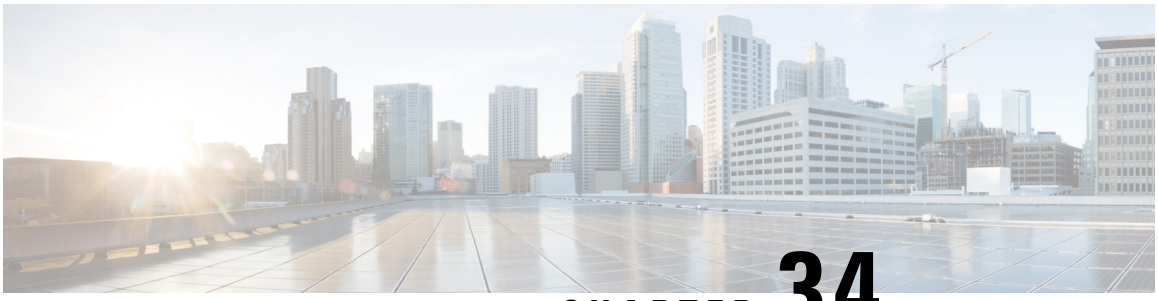
Feature Information for SSH Support over IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 57: Feature Information for SSH Support over IPv6

Feature Name	Releases	Feature Information
SSH Support over IPv6	12.2(8)T 12.2(17a)SX1 12.2(25)SEE 12.2(25)SG 12.2(33)SRA 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	SSH provides support for IPv6 addresses that enable a Cisco device to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport. The following commands were introduced or modified: **ip ssh**, **ssh**.



CHAPTER 34

SNMP over IPv6

Simple Network Management Protocol (SNMP) can be configured over IPv6 transport so that an IPv6 host can perform SNMP queries and receive SNMP notifications from a device running IPv6.

Information About SNMP over IPv6

SNMP over an IPv6 Transport

Simple Network Management Protocol (SNMP) can be configured over IPv6 transport so that an IPv6 host can perform SNMP queries and receive SNMP notifications from a device running IPv6 software. The SNMP agent and related MIBs have been enhanced to support IPv6 addressing. This feature uses the data encryption standard (3DES) and advanced encryption standard (AES) message encryption.

How to Configure SNMP over IPv6

Configuring an SNMP Notification Server over IPv6

Use an SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the device. Optionally, you can specify one or more of the following characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent.
- A MIB view, which defines the subset of all MIB objects accessible to the given community.
- Read and write or read-only permission for the MIB objects accessible to the community.

You can configure one or more community strings. To remove a specific community string, use the **no snmp-server community** command.

The **snmp-server host** command specifies which hosts will receive SNMP notifications, and whether you want the notifications sent as traps or inform requests. The **snmp-server enable traps** command globally enables the production mechanism for the specified notification types (such as Border Gateway Protocol [BGP] traps, config traps, entity traps, and Hot Standby Router Protocol [HSRP] traps).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]
4. **snmp-server engineID remote** {*ipv4-ip-address* | *ipv6-address*} [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string*
5. **snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**context** *context-name*] [**read** *read-view*] [**write** *write-view*] [**notify** *notify-view*] [**access** [**ipv6** *named-access-list*] {*acl-number* | *acl-name*}]
6. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
7. **snmp-server user** *username group-name* [**remote** *host* [**udp-port** *port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** [**ipv6** *nacl*] [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *privpassword*] {*acl-number* | *acl-name*}]
8. **snmp-server enable traps** [*notification-type*] [**vrrp**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server community** *string* [**view** *view-name*] [**ro**	**rw**] [**ipv6** *nacl*] [*access-list-number*] **Example:** `Device(config)# snmp-server community mgr view restricted rw ipv6 mgr2`	Defines the community access string.
Step 4	**snmp-server engineID remote** {*ipv4-ip-address*	*ipv6-address*} [**udp-port** *udp-port-number*] [**vrf** *vrf-name*] *engineid-string* **Example:**	(Optional) Specifies the name of the remote SNMP engine (or copy of SNMP).

	Command or Action	Purpose
	`Device(config)# snmp-server engineID remote 3ffe:b00:c18:1::3/127 remotev6`	
Step 5	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**context** *context-name*] [**read** *read-view*] [**write** *write-view*] [**notify** *notify-view*] [**access** [**ipv6** *named-access-list*] {*acl-number* | *acl-name*}] **Example:** `Device(config)# snmp-server group public v2c access ipv6 public2`	(Optional) Configures a new SNMP group, or a table that maps SNMP users to SNMP views.
Step 6	**snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Device(config)# snmp-server host host1.com 2c vrf trap-vrf`	Specifies the recipient of an SNMP notification operation. • Specifies whether you want the SNMP notifications sent as traps or informs, the version of SNMP to use, the security level of the notifications (for SNMPv3), and the recipient (host) of the notifications.
Step 7	**snmp-server user** *username group-name* [**remote** *host* [**udp-port** *port*]] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**} *auth-password*]} [**access** [**ipv6** *nacl*] [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *privpassword*] {*acl-number* | *acl-name*}] **Example:** `Device(config)# snmp-server user user1 bldg1 remote 3ffe:b00:c18:1::3/127 v2c access ipv6 public2`	(Optional) Configures a new user to an existing SNMP group. **Note** You cannot configure a remote user for an address without first configuring the engine ID for that remote host. This is a restriction imposed in the design of these commands; if you try to configure the user before the host, you will receive a warning message, and the command will not be executed.
Step 8	**snmp-server enable traps** [*notification-type*] [**vrrp**] **Example:** `Device(config)# snmp-server enable traps bgp`	Enables sending of traps or informs, and specifies the type of notifications to be sent. • If a value for the *notification-type* argument is not specified, all supported notification will be enabled on the device. • To discover which notifications are available on your device, enter the **snmp-server enable traps ?** command.

Configuration Examples for SNMP over IPv6

Examples: Configuring an SNMP Notification Server over IPv6

The following example permits any SNMP to access all objects with read-only permission using the community string named public. The device also will send Border Gateway Protocol (BGP) traps to the IPv4 host 172.16.1.111 and IPv6 host 3ffe:b00:c18:1::3/127 using SNMPv1 and to the host 172.16.1.27 using SNMPv2c. The community string named public will be sent with the traps.

```
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps bgp
Device(config)# snmp-server host 172.16.1.27 version 2c public
Device(config)# snmp-server host 172.16.1.111 version 1 public
Device(config)# snmp-server host 3ffe:b00:c18:1::3/127 public
```

Example: Associate an SNMP Server Group with Specified Views

In the following example, the SNMP context A is associated with the views in SNMPv2c group GROUP1 and the IPv6 named access list public2:

```
Device(config)# snmp-server context A
Device(config)# snmp mib community-map commA context A target-list commAVpn
Device(config)# snmp mib target list commAVpn vrf CustomerA
Device(config)# snmp-server view viewA ciscoPingMIB included
Device(config)# snmp-server view viewA ipForward included
Device(config)# snmp-server group GROUP1 v2c context A read viewA write viewA notify
access ipv6 public2
```

Example: Create an SNMP Notification Server

The following example configures the IPv6 host as the notification server:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server community mgr view restricted rw ipv6 mgr2
Device(config)# snmp-server engineID remote 3ffe:b00:c18:1::3/127 remotev6
Device(config)# snmp-server group public v2c access ipv6 public2
Device(config)# snmp-server host host1.com 2c vrf trap-vrf
Device(config)# snmp-server user user1 bldg1 remote 3ffe:b00:c18:1::3/127 v2c access ipv6
public2
Device(config)# snmp-server enable traps bgp
Device(config)# exit
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

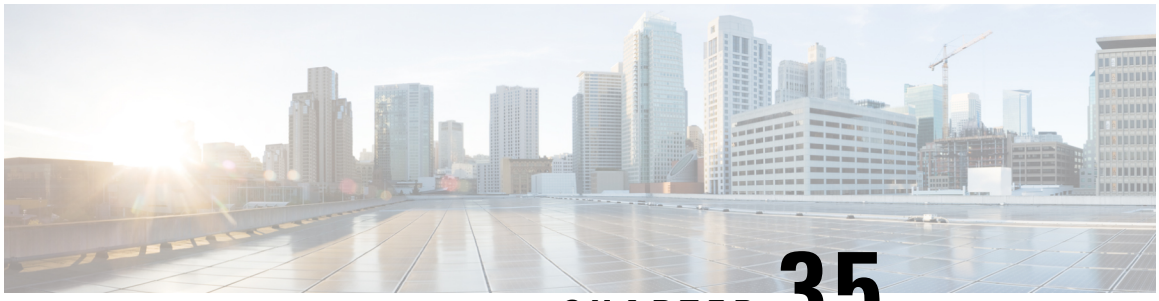
Feature Information for SNMP over IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 58: Feature Information for SNMP over IPv6

Feature Name	Releases	Feature Information
SNMP over IPv6	12.2(33)SRB 12.2(33)SXI 12.2(44)SE 12.2(44)SG 12.3(14)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	SNMP can be configured over IPv6 transport so that an IPv6 host can perform SNMP queries and receive SNMP notifications from a device running IPv6. The following commands were introduced or modified: **snmp-server community**, **snmp-server enable traps**, **snmp-server engineID remote**, **snmp-server group**, **snmp-server host**, **snmp-server user**.
SNMPv3--3DES and AES Encryption Support	12.2(33)SRB 12.2(33)SXI 12.2(50)SG 12.2(52)SE 12.4(2)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports the SNMPv3 - 3DES and AES Encryption Support feature. No commands were introduced or modified.



CHAPTER 35

IPv6 MIBs

This document is about MIBs that are implemented for IPv6. Cisco has long supported IP-MIB and IP-FORWARD-MIB in IPv4. CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB are IPv6 MIBs that are defined as being protocol-independent, but they are implemented only for IPv6 objects and tables.

Information About IPv6 MIBs

Cisco IPv6 MIBs

Cisco has long supported IP-MIB and IP-FORWARD-MIB in IPv4. CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB are IPv6 MIBs that are defined as being protocol-independent, but are implemented only for IPv6 objects and tables. IP-MIB and IP-FORWARD-MIB adhere to RFC 4293 and RFC 4292 standards, as follows:

- The upgrade is backward-compatible; all IP-MIB and IP-FORWARD-MIB objects and tables still appear.
- IP-MIB and IP-FORWARD-MIB include definitions of new IPv6-only, IPv4-only, and protocol-version independent (PVI) objects and tables.

CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB were removed from the Cisco releases in which CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB were applied. Information in CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB is included IP-MIB and IP-FORWARD-MIB.

MIBs Supported for IPv6

The following MIBs are supported for IPv6:

- CISCO-CONFIG-COPY-MIB
- CISCO-CONFIG-MAN-MIB
- CISCO-DATA-COLLECTION-MIB

- CISCO-FLASH-MIB
- CISCO-SNMP-TARGET-EXT-MIB
- ENTITY-MIB
- IP-FORWARD-MIB
- IP-MIB
- NOTIFICATION-LOG-MIB
- SNMP-TARGET-MIB

CISCO-CONFIG-COPY-MIB and CISCO-FLASH-MIB support IPv6 addressing when TFTP, remote copy protocol (rcp), or FTP is used.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 MIBs

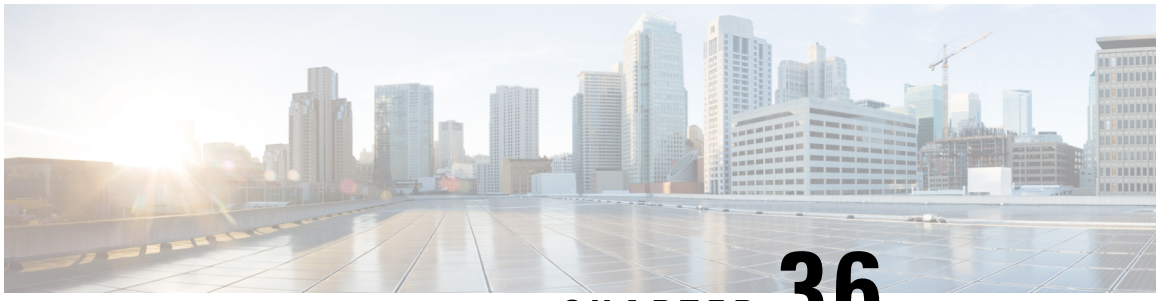
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 59: Feature Information for IPv6 MIBs

Feature Name	Releases	Feature Information
IPv6 MIBs	12.0(22)S 12.2(14)S 12.2(15)T 12.2(28)SB 12.2(33)SRA 12.2(50)SY 15.0(1)SY Cisco IOS XE Release 2.1 Cisco IOS XE Release 3.9S	This feature is supported in IPv6. No commands were introduced or modified. In Cisco IOS XE Release 3.9S, support was added for the Cisco ISR 4400 Series Routers.

Feature Name	Releases	Feature Information
IPv6 Services: RFC 4293 IP-MIB (IPv6 Only) and RFC 4292 IP-FORWARD-MIB (IPv6 Only)	12.2(33)SRC 12.2(50)SY 12.2(54)SG 12.2(58)SE 15.0(2)SG 15.0(1)SY 15.1(3)T Cisco IOS XE Release 2.1 3.2SG	IP-FORWARD-MIB and IP-MIB were updated to RFC 4292 and RFC 4293 standards, respectively. No commands were introduced or modified.



CHAPTER 36

IPv6 Embedded Management Components

Cisco IPv6 embedded management components have IPv6-compliant operability in IPv6 and hybrid IPv6 and IPv4 networks. This document describes the following embedded management components: syslog, config logger, TCL, NETCONF, and the SOAP message format.

Information About IPv6 Embedded Management Components

Syslog

The Cisco system message logging (syslog) process in IPv6 allows users to log syslog messages to external syslog servers and hosts with IPv6 addresses. This implementation allows user to specify an IPv4-based logging host (syslog server) by providing the host's IP address in IPv4 format (for example, 192.168.0.0) or IPv6 format (for example, 2001:DB8:A00:1::1/64).

Config Logger

Config logger tracks and reports configuration changes. Config logger supports two content types:

- Plain text--With plain-text format, the config logger reports configuration changes only.
- XML--The config logger uses XML to report the configuration change details (for example, what changed, who changed it, when changes were made, parser return code [PRC] values, and incremental NVGEN results).

TCL

Tool command language (TCL) is used in Cisco software for IPv6 to support features such as embedded syslog manager (ESM), embedded event manager (EEM), interactive voice response (IVR), and tclsh parser mode. TCL supports both initiating (client) and listening (server) sockets.

NETCONF

The Network Configuration Protocol (NETCONF) defines a mechanism through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. NETCONF uses XML-based data encoding for the configuration data and protocol messages.

SOAP Message Format

Using the Service-Oriented Access Protocol (SOAP) provides a way to format the layout of Cisco Networking Services (CNS) messages in a consistent manner. SOAP is intended for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework that provides a message format that can be exchanged over a variety of underlying protocols.

Within the SOAP message structure, there is a security header that enables CNS notification messages to authenticate user credentials.

How to Configure IPv6 Embedded Management Components

Configuring Syslog over IPv6

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **logging host** {{*ip-address* | *hostname*} | {**ipv6** *ipv6-address* | *hostname*}} [**transport** {**udp** [**port** *port-number*] | **tcp** [**port** *port-number*] [**audit**]}] [**xml** | **filtered** [**stream** *stream-id*]] [**alarm** [*severity*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**logging host** {{*ip-address* | *hostname*} | {**ipv6** *ipv6-address* | *hostname*}} [**transport** {**udp** [**port** *port-number*] | **tcp** [**port** *port-number*] [**audit**]}] [**xml** | **filtered** [**stream** *stream-id*]] [**alarm** [*severity*]] **Example:**	Logs system messages and debug output to a remote host.

	Command or Action	Purpose
	`Device(config)# logging host ipv6 AAAA:BBBB:CCCC:DDDD::FFFF`	

Configuration Examples for IPv6 Embedded Management Components

Example: Configuring Syslog over IPv6

```
Device(config)# logging host ipv6 AAAA:BBBB:CCCC:DDDD::FFFF transport tcp port 1470
```

Additional References for IPv6 Embedded Management Components

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco_IOS_IPv6_Feature_Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Embedded Management Components

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

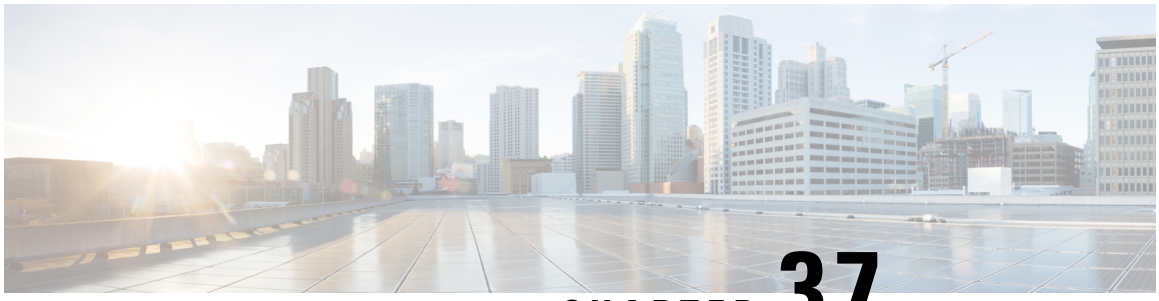
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 60: Feature Information for IPv6 Embedded Management Components

Feature Name	Releases	Feature Information
IPv6: Config Logger	12.2(33)SB 12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(1)SY 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.

Feature Name	Releases	Feature Information
IPv6: NETCONF	12.2(33)SB 12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.
IPv6 Support in SOAP	12.2(33)SB 12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.
IPv6: TCL	12.2(33)SB 12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(1)SY 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.

Feature Name	Releases	Feature Information
Syslog over IPv6	12.2(33)SB 12.2(33)SRC 12.2(33)SXI 12.2(44)SE 12.2(44)SG 12.4(4)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	The Cisco syslog process in IPv6 allows users to log syslog messages to external syslog servers and hosts with IPv6 addresses. The following command was introduced: **logging host**.



CHAPTER 37

IPv6 CNS Agents

IPv6 addressing is supported in the Cisco Networking Services (CNS) subsystem. CNS is a foundation technology for linking users to networking services and provides the infrastructure for the automated configuration of large numbers of network devices. The document describes CNS agents supported in IPv6.

Information About IPv6 CNS Agents

CNS Agents

IPv6 addressing is supported in the Cisco Networking Services (CNS) subsystem. CNS is a foundation technology for linking users to networking services, and it provides the infrastructure for the automated configuration of large numbers of network devices. Many IPv6 networks are complex, with many devices, and each device must be configured individually. When standard configurations do not exist or have been modified, the time involved in initial installation and subsequent upgrading is considerable. ISPs need a method for sending out partial configurations to introduce new services.

To address all these issues, CNS was designed to provide "plug-and-play" network services using a central directory service and distributed agents. CNS features include CNS agents and a flow-through provisioning structure. CNS flow-through provisioning uses the CNS configuration and event agents to provide an automated workflow, eliminating the need for an onsite technician.

IPv6 addressing supports the CNS agents described in the following sections:

CNS Configuration Agent

The CNS configuration agent is involved in the initial configuration and subsequent partial configurations on a Cisco device. The configuration agent uses a CNS configuration engine to provide methods for automating initial Cisco device configurations, incremental configurations, and synchronized configuration updates, and the configuration engine reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe.

CNS Event Agent

The CNS event agent provides a transport connection to the CNS event bus for all other CNS agents. No event can be sent to the device by the configuration engine until the CNS event agent is operational and has successfully built a connection between the configuration engine and the device.

The event agent uses a CNS configuration engine to provide methods for automating initial Cisco device configurations, incremental configurations, and synchronized configuration updates.

CNS EXEC Agent

The CNS EXEC agent allows a remote application to execute a CLI command in EXEC mode on a Cisco device by sending an event message that contains the command.

CNS Image Agent

Administrators maintaining large networks of Cisco devices need an automated mechanism to load image files onto large numbers of remote devices. Network management applications are useful to determine which images to run and how to manage images received from the Cisco online software center. Other image distribution solutions do not scale to cover thousands of devices and cannot distribute images to devices behind a firewall or using Network Address Translation (NAT). The CNS image agent enables the managed device to initiate a network connection and request an image download allowing devices using NAT, or behind firewalls, to access the image server.

The CNS image agent can be configured to use the CNS event bus. To use the CNS event bus, the CNS event agent must be enabled and connected to the CNS event gateway in the CNS Configuration Engine. The CNS image agent can also use an HTTP server that understands the CNS image agent protocol. Deployment of CNS image agent operations can use both the CNS event bus and an HTTP server.

Additional References for IPv6 IOS Firewall

Related Documents

Related Topic	Document Title
Security commands	• *Cisco IOS Security Command Reference: Commands A to C* • *Cisco IOS Security Command Reference: Commands D to L* • *Cisco IOS Security Command Reference: Commands M to R* • *Cisco IOS Security Command Reference: Commands S to Z*
IPv6 commands	*Cisco IOS IPv6 Command Reference*
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS IPv6 features	*Cisco IOS IPv6 Feature Mapping*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

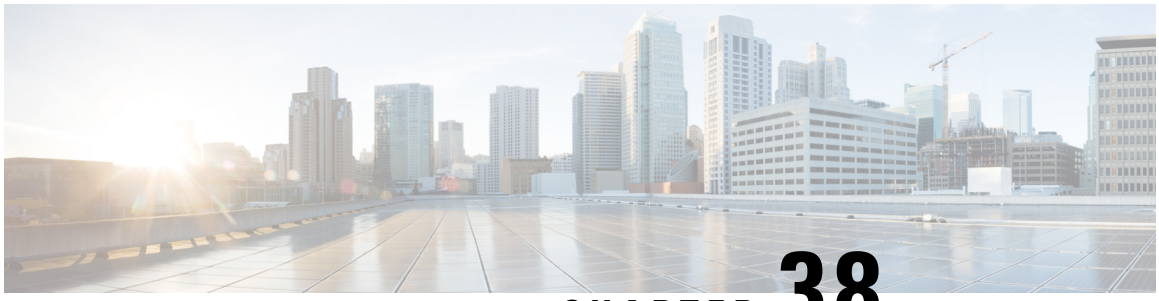
Feature Information for IPv6 CNS Agents

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 61: Feature Information for IPv6 CNS Agents

Feature Name	Releases	Feature Information
IPv6 CNS Agents	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T Cisco IOS XE Release 3.9S	CNS configuration and event agents use a CNS configuration engine to provide methods for automating initial device configurations, incremental configurations, and synchronized configuration updates, and the configuration engine reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe. No commands were introduced or modified. In Cisco IOS XE Release 3.9S, support was added for the Cisco CSR 1000V.



CHAPTER 38

IPv6 HTTP(S)

Hypertext Transfer Protocol server HTTP(S) is a Cisco IPv6 embedded management component. Cisco IPv6 embedded management components have IPv6-compliant operability in IPv6 and hybrid IPv6 and IPv4 networks.

Information About IPv6 HTTP(S)

Cisco IPv6 Embedded Management Components

Cisco embedded management components have IPv6-compliant operability in IPv6 and dual-stack IPv6 and IPv4 networks.

HTTP(S) IPv6 Support

This feature allows the HTTP(S) client and server to support IPv6 addresses.

The HTTP server in Cisco software can service requests from both IPv6 and IPv4 HTTP clients. When the HTTP(S) server accepts a connection from a client, the server determines whether the client is an IPv4 or IPv6 host. The address family, IPv4 or IPv6, for the accept socket call is then chosen accordingly. The listening socket continues to listen for both IPv4 and IPv6 connections.

The HTTP client in Cisco software can send requests to both IPv4 and IPv6 HTTP servers.

When you use the IPv6 HTTP client, URLs with literal IPv6 addresses must be formatted using the rules listed in RFC 2732.

How to Configure IPv6 HTTP(S)

Disabling HTTP Access to an IPv6 Device

HTTP access over IPv6 is automatically enabled if an HTTP server is enabled and the device has an IPv6 address. If the HTTP server is not required, it should be disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip http server**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip http server** **Example:** `Device(config)# no ip http server`	Disables HTTP access.

Configuration Examples for IPv6 HTTP(S)

Example: Disabling HTTP Access to the Device

In the following example, the **show running-config** command is used to show that HTTP access is disabled on the device:

```
Device# show running-config

Building configuration...
!
Current configuration : 1490 bytes
!
version 12.2
```

```
!
hostname Device
!
no ip http server
!
line con 0
line aux 0
line vty 0 4
```

Additional References

Related Documents

Related Topic	Document Title
IP access list commands	*Cisco IOS Security Command Reference*
Configuring IP access lists	*Creating an IP Access List and Applying It to an Interface*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

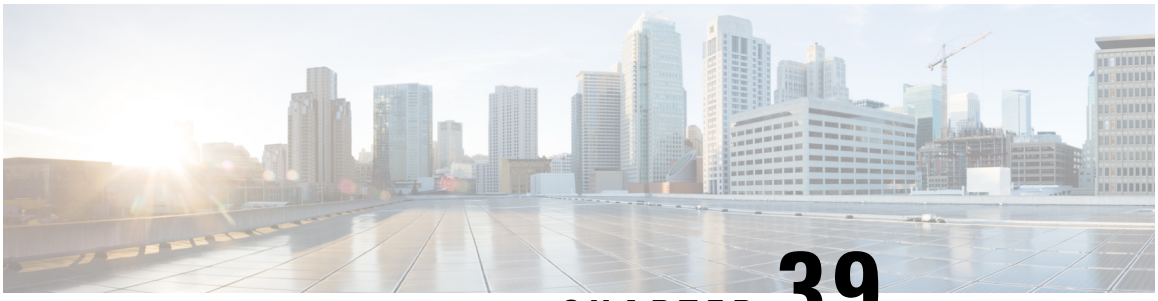
Feature Information for IPv6 HTTP(S)

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 62: Feature Information for IPv6 HTTP(S)

Feature Name	Releases	Feature Information
IPv6 HTTP(S)	12.2(33)SB 12.2(33)SRC 12.2(50)SY 12.4(20)T 15.0(1)SY Cisco IOS XE Release 3.8S	This feature enables the HTTP(S) client and server to support IPv6 addresses. The following command was modified: **ip http server**.



CHAPTER 39

IP SLAs for IPv6

Cisco IP Service Level Agreements (SLAs) are a portfolio of technology embedded in most devices that run Cisco software. SLAs allow Cisco customers to analyze IPv6 service levels for IPv6 applications and services, increase productivity, lower operational costs, and reduce the frequency of network outages.

Information About IP SLAs for IPv6

Cisco IPv6 Embedded Management Components

Cisco embedded management components have IPv6-compliant operability in IPv6 and dual-stack IPv6 and IPv4 networks.

IP SLAs for IPv6

Cisco IP Service Level Agreements (SLAs) are a portfolio of technology embedded in most devices that run Cisco software that allows Cisco customers to analyze IPv6 service levels for IPv6 applications and services, increase productivity, lower operational costs, and reduce the frequency of network outages. IP SLAs uses active traffic monitoring--the generation of traffic in a continuous, reliable, and predictable manner--for measuring network performance.

The following Cisco IP SLAs are supported for IPv6:

- Internet Control Message Protocol (ICMP) echo operation--Used to monitor end-to-end response time between a Cisco device and other devices using IPv4 or IPv6. ICMP echo is useful for troubleshooting network connectivity issues.
- TCP connect operation--Used to measure the response time taken to perform a TCP Connect operation between a Cisco device and other devices using IPv4 or IPv6.
- User Datagram Protocol (UDP) echo operation--Used to monitor end-to-end response time between a Cisco router and devices using IPv4 or IPv6 .
- UDP jitter operation--Used to analyze round-trip delay, one-way delay, one-way jitter, one-way packet loss, and connectivity in networks that carry UDP traffic in IPv4 or IPv6 networks.

- UDP jitter operation--Used to monitor VoIP quality levels in your network, allowing you to guarantee VoIP quality levels to your users in IPv4 or IPv6 networks.

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IPv6 commands	*Cisco IOS IPv6 Command Reference*
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

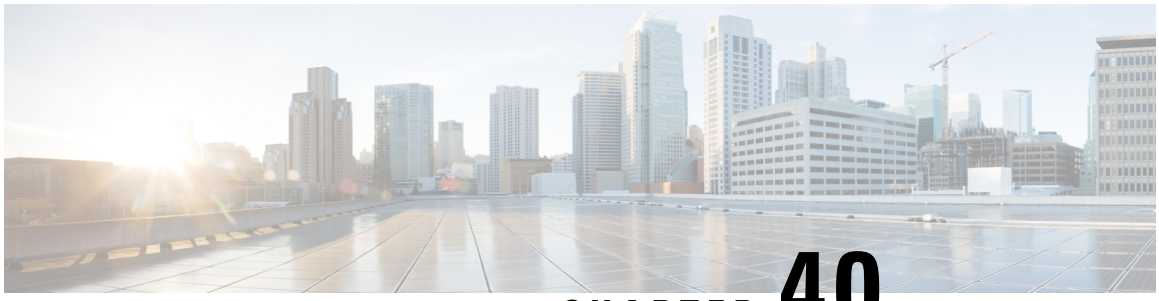
Feature Information for IP SLAs for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 63: Feature Information for IP SLAs for IPv6

Feature Name	Releases	Feature Information
IP SLAs for IPv6	12.2(33)SRC 12.2(50)SG 12.2(50)SY 12.4(20)T 15.0(2)SG Cisco IOS XE Release 2.1 3.2SG	IPv6 supports this feature. No commands were introduced or modified.



CHAPTER 40

IPv6 RFCs

Standards and RFCs

RFCs	Title
RFC 1195	*Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*
RFC 1267	*A Border Gateway Protocol 3 (BGP-3)*
RFC 1305	*Network Time Protocol (Version 3) Specification, Implementation and Analysis*
RFC 1583	*OSPF version 2*
RFC 1772	*Application of the Border Gateway Protocol in the Internet*
RFC 1886	*DNS Extensions to Support IP version 6*
RFC 1918	*Address Allocation for Private Internets*
RFC 1981	*Path MTU Discovery for IP version 6*
RFC 2080	*RIPng for IPv6*
RFC 2281	*Cisco Hot Standby Router Protocol (HSRP)*
RFC 2332	*NBMA Next Hop Resolution Protocol (NHRP)*
RFC 2373	*IP Version 6 Addressing Architecture*
RFC 2374	*An Aggregatable Global Unicast Address Format*
RFC 2375	*IPv6 Multicast Address Assignments*
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2402	*IP Authentication Header*
RFC 2404	*The Use of Hash Message Authentication Code Federal Information Processing Standard 180-1 within Encapsulating Security Payload and Authentication Header*
RFC 2406	*IP Encapsulating Security Payload (ESP)*

RFCs	Title
RFC 2407	*The Internet Security Domain of Interpretation for ISAKMP*
RFC 2408	*Internet Security Association and Key Management Protocol*
RFC 2409	*Internet Key Exchange (IKE)*
RFC 2427	*Multiprotocol Interconnect over Frame Relay*
RFC 2428	*FTP Extensions for IPv6 and NATs*
RFC 2460	*Internet Protocol, Version 6 (IPv6) Specification*
RFC 2461	*Neighbor Discovery for IP Version 6 (IPv6)*
RFC 2462	*IPv6 Stateless Address Autoconfiguration*
RFC 2463	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 2464	*Transmission of IPv6 Packets over Ethernet*
RFC 2467	*Transmission of IPv6 Packets over FDDI*
RFC 2472	*IP Version 6 over PPP*
RFC 2473	*Generic Packet Tunneling in IPv6 Specification*
RFC 2474	*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*
RFC 2475	*An Architecture for Differentiated Services Framework*
RFC 2492	*IPv6 over ATM*
RFC 2545	*Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*
RFC 2590	*Transmission of IPv6 Packets over Frame Relay Specification*
RFC 2597	*Assured Forwarding PHB*
RFC 2598	*An Expedited Forwarding PHB*
RFC 2640	*Internet Protocol, Version 6 Specification*
RFC 2684	*Multiprotocol Encapsulation over ATM Adaptation Layer 5*
RFC 2697	*A Single Rate Three Color Marker*
RFC 2698	*A Two Rate Three Color Marker*
RFC 2710	*Multicast Listener Discovery (MLD) for IPv6*
RFC 2711	*IPv6 Router Alert Option*
RFC 2732	*Format for Literal IPv6 Addresses in URLs*

RFCs	Title
RFC 2765	*Stateless IP/ICMP Translation Algorithm (SIIT)*
RFC 2766	*Network Address Translation-Protocol Translation (NAT-PT)*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 2893	*Transition Mechanisms for IPv6 Hosts and Routers*
RFC 3056	*Connection of IPv6 Domains via IPv4 Clouds*
RFC 3068	*An Anycast Prefix for 6to4 Relay Routers*
RFC 3095	*RObust Header Compression (ROHC): Framework and Four Profiles: RTP, UDP, ESP, and Uncompressed*
RFC 3107	*Carrying Label Information in BGP-4*
RFC 3137	*OSPF Stub Router Advertisement*
RFC 3147	*Generic Routing Encapsulation over CLNS*
RFC 3152	*Delegation of IP6.ARPA*
RFC 3162	*RADIUS and IPv6*
RFC 3315	*Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3319	*Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiated Protocol (SIP) Servers*
RFC 3392	*Capabilities Advertisement with BGP-4*
RFC 3414	*User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3484	*Default Address Selection for Internet Protocol version 6 (IPv6)*
RFC 3513	*Internet Protocol Version 6 (IPv6) Addressing Architecture*
RFC 3576	*Change of Authorization*
RFC 3587	*IPv6 Global Unicast Address Format*
RFC 3590	*Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*
RFC 3596	*DNS Extensions to Support IP Version 6*
RFC 3633	*DHCP IPv6 Prefix Delegation*
RFC 3646	*DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3697	*IPv6 Flow Label Specification*
RFC 3736	*Stateless DHCP Service for IPv6*

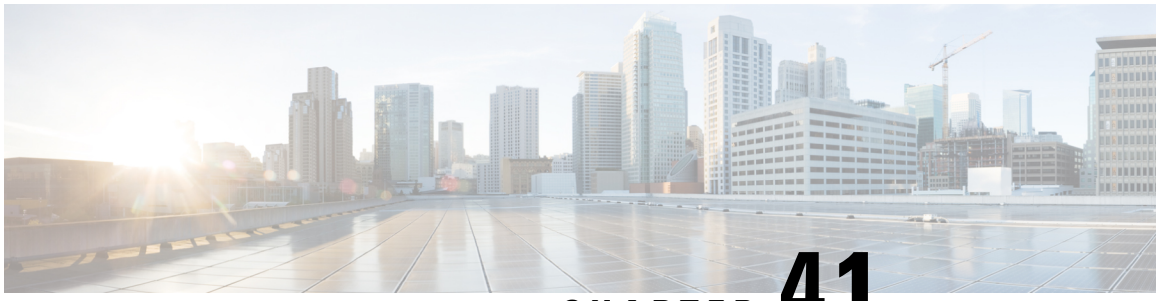
RFCs	Title
RFC 3756	*IPv6 Neighbor Discovery (ND) Trust Models and Threats*
RFC 3759	*RObust Header Compression (ROHC): Terminology and Channel Mapping Examples*
RFC 3775	*Mobility Support in IPv6*
RFC 3810	*Multicast Listener Discovery Version 2 (MLDv2) for IPv6*
RFC 3846	*Mobile IPv4 Extension for Carrying Network Access Identifiers*
RFC 3879	*Deprecating Site Local Addresses*
RFC 3898	*Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 3954	*Cisco Systems NetFlow Services Export Version 9*
RFC 3956	*Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*
RFC 3963	*Network Mobility (NEMO) Basic Support Protocol*
RFC 3971	*SEcure Neighbor Discovery (SEND)*
RFC 3972	*Cryptographically Generated Addresses (CGA)*
RFC 4007	*IPv6 Scoped Address Architecture*
RFC 4075	*Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6*
RFC 4087	*IP Tunnel MIB*
RFC 4091	*The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework*
RFC 4092	*Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)*
RFC 4109	*Algorithms for Internet Key Exchange version 1 (IKEv1)*
RFC 4191	*Default Router Preferences and More-Specific Routes*
RFC 4193	*Unique Local IPv6 Unicast Addresses*
RFC 4214	*Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*
RFC 4242	*Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*
RFC 4282	*The Network Access Identifier*
RFC 4283	*Mobile Node Identifier Option for Mobile IPv6*
RFC 4285	*Authentication Protocol for Mobile IPv6*
RFC 4291	*IP Version 6 Addressing Architecture*

RFCs	Title
RFC 4292	*IP Forwarding Table MIB*
RFC 4293	*Management Information Base for the Internet Protocol (IP)*
RFC 4302	*IP Authentication Header*
RFC 4306	*Internet Key Exchange (IKEv2) Protocol*
RFC 4308	*Cryptographic Suites for IPsec*
RFC 4364	*BGP MPLS/IP Virtual Private Networks (VPNs)*
RFC 4382	*MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base*
RFC 4443	*Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*
RFC 4552	*Authentication/Confidentiality for OSPFv3*
RFC 4594	*Configuration Guidelines for DiffServ Service Classes*
RFC 4601	*Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification*
RFC 4610	*Anycast-RP Using Protocol Independent Multicast (PIM)*
RFC 4649	*Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option*
RFC 4659	*BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*
RFC 4724	*Graceful Restart Mechanism for BGP*
RFC 4798	*Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*
RFC 4818	*RADIUS Delegated-IPv6-Prefix Attribute*
RFC 4861	*Neighbor Discovery for IP version 6 (IPv6)*
RFC 4862	*IPv6 Stateless Address Autoconfiguration*
RFC 4884	*Extended ICMP to Support Multi-Part Messages*
RFC 4885	*Network Mobility Support Terminology*
RFC 4887	*Network Mobility Home Network Models*
RFC 5015	*Bidirectional Protocol Independent Multicast (BIDIR-PIM)*
RFC 5059	*Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*
RFC 5072	*IPv6 over PPP*
RFC 5095	*Deprecation of Type 0 Routing Headers in IPv6*
RFC 5120	*M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)*

RFCs	Title
RFC 5130	*A Policy Control Mechanism in IS-IS Using Administrative Tags*
RFC 5187	*OSPFv3 Graceful Restart*
RFC 5213	*Proxy Mobile IPv6*
RFC 5308	*Routing IPv6 with IS-IS*
RFC 5340	*OSPF for IPv6*
RFC 5460	*DHCPv6 Bulk Leasequery*
RFC 5643	*Management Information Base for OSPFv3*
RFC 5838	*Support of Address Families in OSPFv3*
RFC 5844	*IPv4 Support for Proxy Mobile IPv6*
RFC 5845	*Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6*
RFC 5846	*Binding Revocation for IPv6 Mobility*
RFC 5881	*Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)*
RFC 5905	*Network Time Protocol Version 4: Protocol and Algorithms Specification*
RFC 5969	*IPv6 Rapid Deployment on IPv4 Infrastructures (6RD) -- Protocol Specification*
RFC 6105	*IPv6 Router Advertisement Guard*
RFC 6620	*FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses*

PART III

First Hop Redundancy Protocols



CHAPTER 41

Configuring GLBP

Gateway Load Balancing Protocol (GLBP) protects data traffic from a failed device or circuit, like Hot Standby Router Protocol (HSRP) and Virtual Router Redundancy Protocol (VRRP), while allowing packet load sharing between a group of redundant devices.

Restrictions for GLBP

Enhanced Object Tracking (EOT) is not stateful switchover (SSO)-aware and cannot be used with GLBP in SSO mode.

Prerequisites for GLBP

Before configuring GLBP, ensure that the devices can support multiple MAC addresses on the physical interfaces. For each GLBP forwarder to be configured, an additional MAC address is used.

Information About GLBP

GLBP Overview

GLBP provides automatic device backup for IP hosts configured with a single default gateway on an IEEE 802.3 LAN. Multiple first-hop devices on the LAN combine to offer a single virtual first-hop IP device while sharing the IP packet forwarding load. Other devices on the LAN act as redundant GLBP devices that will become active if any of the existing forwarding devices fail.

GLBP performs a similar function for the user as HSRP and VRRP. HSRP and VRRP allow multiple devices to participate in a virtual device group configured with a virtual IP address. One member is elected to be the active device to forward packets sent to the virtual IP address for the group. The other devices in the group are redundant until the active device fails. These standby devices have unused bandwidth that the protocol is not using. Although multiple virtual device groups can be configured for the same set of devices, the hosts must be configured for different default gateways, which results in an extra administrative burden. The advantage of GLBP is that it additionally provides load balancing over multiple devices (gateways) using a single virtual IP address and multiple virtual MAC addresses. The forwarding load is shared among all devices in a GLBP group rather than being handled by a single device while the other devices stand idle. Each host is configured with the same virtual IP address, and all devices in the virtual device group participate in forwarding packets. GLBP members communicate between each other through hello messages sent every 3 seconds to the multicast address 224.0.0.102, UDP port 3222 (source and destination).

GLBP Packet Types

GLBP uses 3 different packet types to operate. The packet types are Hello, Request, and Reply. The Hello packet is used to advertise protocol information. Hello packets are multicast, and are sent when any virtual gateway or virtual forwarder is in Speak, Standby or Active state. Request and Reply packets are used for virtual MAC assignment. They are both unicast messages to and from the active virtual gateway (AVG).

GLBP Active Virtual Gateway

Members of a GLBP group elect one gateway to be the active virtual gateway (AVG) for that group. Other group members provide backup for the AVG if the AVG becomes unavailable. The AVG assigns a virtual MAC address to each member of the GLBP group. Each gateway assumes responsibility for forwarding packets sent to the virtual MAC address assigned to it by the AVG. These gateways are known as active virtual forwarders (AVFs) for their virtual MAC address.

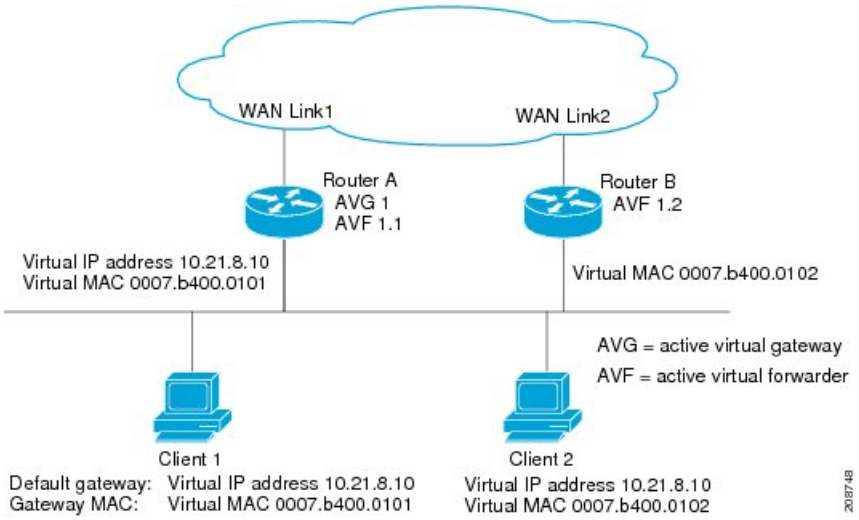
The AVG is also responsible for answering Address Resolution Protocol (ARP) requests for the virtual IP address. Load sharing is achieved by the AVG replying to the ARP requests with different virtual MAC addresses.

Prior to Cisco IOS Release 15.0(1)M1 and 12.4(24)T2, when the **no glbp load-balancing** command is configured, the AVG always responds to ARP requests with the MAC address of its AVF.

In Cisco IOS Release 15.0(1)M1 and 12.4(24)T2, and later releases, when the **no glbp load-balancing** command is configured, if the AVG does not have an AVF, it preferentially responds to ARP requests with the MAC address of the first listening virtual forwarder (VF), which will causes traffic to route via another gateway until that VF migrates back to being the current AVG.

In the figure below, Router A (or Device A) is the AVG for a GLBP group, and is responsible for the virtual IP address 10.21.8.10. Router A is also an AVF for the virtual MAC address 0007.b400.0101. Router B (or Device B) is a member of the same GLBP group and is designated as the AVF for the virtual MAC address 0007.b400.0102. Client 1 has a default gateway IP address of 10.21.8.10 and a gateway MAC address of 0007.b400.0101. Client 2 shares the same default gateway IP address but receives the gateway MAC address 0007.b400.0102 because Router B is sharing the traffic load with Router A.

Figure 9: GLBP Topology



If Router A becomes unavailable, Client 1 will not lose access to the WAN because Router B will assume responsibility for forwarding packets sent to the virtual MAC address of Router A, and for responding to packets sent to its own virtual MAC address. Router B will also assume the role of the AVG for the entire GLBP group. Communication for the GLBP members continues despite the failure of a device in the GLBP group.

GLBP Virtual MAC Address Assignment

A GLBP group allows up to four virtual MAC addresses per group. The AVG is responsible for assigning the virtual MAC addresses to each member of the group. Other group members request a virtual MAC address after they discover the AVG through hello messages. Gateways are assigned the next MAC address in sequence. A virtual forwarder that is assigned a virtual MAC address by the AVG is known as a primary virtual forwarder. Other members of the GLBP group learn the virtual MAC addresses from hello messages. A virtual forwarder that has learned the virtual MAC address is referred to as a secondary virtual forwarder.

GLBP Virtual Gateway Redundancy

GLBP operates virtual gateway redundancy in the same way as HSRP. One gateway is elected as the AVG, another gateway is elected as the standby virtual gateway, and the remaining gateways are placed in a listen state.

If an AVG fails, the standby virtual gateway will assume responsibility for the virtual IP address. A new standby virtual gateway is then elected from the gateways in the listen state.

GLBP Virtual Forwarder Redundancy

Virtual forwarder redundancy is similar to virtual gateway redundancy with an AVF. If the AVF fails, one of the secondary virtual forwarders in the listen state assumes responsibility for the virtual MAC address.

The new AVF is also a primary virtual forwarder for a different forwarder number. GLBP migrates hosts away from the old forwarder number using two timers that start as soon as the gateway changes to the active virtual forwarder state. GLBP uses the hello messages to communicate the current state of the timers.

The redirect time is the interval during which the AVG continues to redirect hosts to the old virtual forwarder MAC address. When the redirect time expires, the AVG stops using the old virtual forwarder MAC address in ARP replies, although the virtual forwarder will continue to forward packets that were sent to the old virtual forwarder MAC address.

The secondary holdtime is the interval during which the virtual forwarder is valid. When the secondary holdtime expires, the virtual forwarder is removed from all gateways in the GLBP group. The expired virtual forwarder number becomes eligible for reassignment by the AVG.

GLBP Gateway Priority

GLBP gateway priority determines the role that each GLBP gateway plays and what happens if the AVG fails.

Priority also determines if a GLBP device functions as a backup virtual gateway and the order of ascendancy to becoming an AVG if the current AVG fails. You can configure the priority of each backup virtual gateway with a value of 1 through 255 using the **glbp priority** command.

In the "GLBP Topology" figure, if Router A (or Device A)—the AVG in a LAN topology—fails, an election process takes place to determine which backup virtual gateway should take over. In this example, Router B (or Device B) is the only other member in the group so it will automatically become the new AVG. If another device existed in the same GLBP group with a higher priority, then the device with the higher priority would be elected. If both devices have the same priority, the backup virtual gateway with the higher IP address would be elected to become the active virtual gateway.

By default, the GLBP virtual gateway preemptive scheme is disabled. A backup virtual gateway can become the AVG only if the current AVG fails, regardless of the priorities assigned to the virtual gateways. You can enable the GLBP virtual gateway preemptive scheme using the **glbp preempt** command. Preemption allows a backup virtual gateway to become the AVG, if the backup virtual gateway is assigned a higher priority than the current AVG.

GLBP Gateway Weighting and Tracking

GLBP uses a weighting scheme to determine the forwarding capacity of each device in the GLBP group. The weighting assigned to a device in the GLBP group can be used to determine whether it will forward packets and, if so, the proportion of hosts in the LAN for which it will forward packets. Thresholds can be set to disable forwarding when the weighting for a GLBP group falls below a certain value, and when it rises above another threshold, forwarding is automatically reenabled.

The GLBP group weighting can be automatically adjusted by tracking the state of an interface within the device. If a tracked interface goes down, the GLBP group weighting is reduced by a specified value. Different interfaces can be tracked to decrement the GLBP weighting by varying amounts.

By default, the GLBP virtual forwarder preemptive scheme is enabled with a delay of 30 seconds. A backup virtual forwarder can become the AVF if the current AVF weighting falls below the low weighting threshold for 30 seconds. You can disable the GLBP forwarder preemptive scheme using the **no glbp forwarder preempt** command or change the delay using the **glbp forwarder preempt delay minimum** command.

GLBP MD5 Authentication

GLBP MD5 authentication uses the industry-standard MD5 algorithm for improved reliability and security. MD5 authentication provides greater security than the alternative plain text authentication scheme and protects against spoofing software.

MD5 authentication allows each GLBP group member to use a secret key to generate a keyed MD5 hash that is part of the outgoing packet. A keyed hash of an incoming packet is generated and, if the hash within the incoming packet does not match the generated hash, the packet is ignored.

The key for the MD5 hash can either be given directly in the configuration using a key string or supplied indirectly through a key chain. The key string cannot exceed 100 characters in length.

A device will ignore incoming GLBP packets from devices that do not have the same authentication configuration for a GLBP group. GLBP has three authentication schemes:

- No authentication
- Plain text authentication
- MD5 authentication

GLBP packets will be rejected in any of the following cases:

- The authentication schemes differ on the device and in the incoming packet.
- MD5 digests differ on the device and in the incoming packet.
- Text authentication strings differ on the device and in the incoming packet.

ISSU-GLBP

GLBP supports In Service Software Upgrade (ISSU). ISSU allows a high-availability (HA) system to run in Stateful Switchover (SSO) mode even when different versions of Cisco IOS software are running on the active and standby Route Processors (RPs) or line cards.

ISSU provides the ability to upgrade or downgrade from one supported Cisco IOS release to another while continuing to forward packets and maintain sessions, thereby reducing planned outage time. The ability to upgrade or downgrade is achieved by running different software versions on the active RP and standby RP for a short period of time to maintain state information between RPs. This feature allows the system to switch over to a secondary RP running upgraded (or downgraded) software and continue forwarding packets without session loss and with minimal or no packet loss. This feature is enabled by default.

For detailed information about ISSU, see the *Cisco IOS In Service Software Upgrade Process* in the *Cisco IOS High Availability Configuration Guide*

For detailed information about ISSU on the 7600 series devices, see the *ISSU and eFSU on Cisco 7600 Series Routers* document.

GLBP SSO

With the introduction of the GLBP SSO functionality, GLBP is stateful switchover (SSO) aware. GLBP can detect when a device is failing over to the secondary router processor (RP) and continue in its current group state.

SSO functions in networking devices (usually edge devices) that support dual RPs. SSO provides RP redundancy by establishing one of the RPs as the active processor and the other RP as the standby processor. SSO also synchronizes critical state information between the RPs so that network state information is dynamically maintained between RPs.

Without SSO-awareness, if GLBP is deployed on a device with redundant RPs, a switchover of roles between the active RP and the standby RP results in the device relinquishing its activity as a GLBP group member and then rejoining the group as if it had been reloaded. The GLBP SSO feature enables GLBP to continue its activities as a group member during a switchover. GLBP state information between redundant RPs is maintained so that the standby RP can continue the device's activities within the GLBP during and after a switchover.

This feature is enabled by default. To disable this feature, use the **no glbp sso** command in global configuration mode.

For more information, see the *Stateful Swithover* document in the *Cisco IOS High Availability Configuration Guide*.

GLBP Benefits

Load Sharing

You can configure GLBP in such a way that traffic from LAN clients can be shared by multiple devices, thereby sharing the traffic load more equitably among available devices.

Multiple Virtual Devices

GLBP supports up to 1024 virtual devices (GLBP groups) on each physical interface of a device and up to four virtual forwarders per group.

Preemption

The redundancy scheme of GLBP enables you to preempt an active virtual gateway (AVG) with a higher priority backup virtual gateway that has become available. Forwarder preemption works in a similar way, except that forwarder preemption uses weighting instead of priority and is enabled by default.

Authentication

GLBP supports the industry-standard message digest 5 (MD5) algorithm for improved reliability, security, and protection against GLBP-spoofing software. A device within a GLBP group with a different authentication string than other devices will be ignored by other group members. You can alternatively use a simple text password authentication scheme between GLBP group members to detect configuration errors.

How to Configure GLBP

Enabling and Verifying GLBP

Perform this task to enable GLBP on an interface and verify its configuration and operation. GLBP is designed to be easy to configure. Each gateway in a GLBP group must be configured with the same group number, and at least one gateway in the GLBP group must be configured with the virtual IP address to be used by the group. All other required parameters can be learned.

Before you begin

If VLANs are in use on an interface, the GLBP group number must be different for each VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **glbp** *group* **ip** [*ip-address* [**secondary**]]
6. **exit**
7. **show glbp** [*interface-type interface-number*] [*group*] [*state*] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.21.8.32 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**glbp** *group* **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# glbp 10 ip 10.21.8.10`	Enables GLBP on an interface and identifies the primary IP address of the virtual gateway. • After you identify a primary IP address, you can use the **glbp** *group* **ip** command again with the **secondary** keyword to indicate additional IP addresses supported by this group.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode, and returns the device to global configuration mode.

	Command or Action	Purpose
Step 7	**show glbp** [*interface-type interface-number*] [*group*] [*state*] [**brief**] **Example:** `Device(config)# show glbp 10`	(Optional) Displays information about GLBP groups on a device. • Use the optional **brief** keyword to display a single line of information about each virtual gateway or virtual forwarder.

Example

In the following example, sample output is displayed about the status of the GLBP group, named 10, on the device:

```
Device# show glbp 10

GigabitEthernet0/0/0 - Group 10
  State is Active
    2 state changes, last state change 23:50:33
  Virtual IP address is 10.21.8.10
  Hello time 5 sec, hold time 18 sec
    Next hello sent in 4.300 secs
  Redirect time 600 sec, forwarder time-out 7200 sec
  Authentication text "stringabc"
  Preemption enabled, min delay 60 sec
  Active is local
  Standby is unknown
  Priority 254 (configured)
  Weighting 105 (configured 110), thresholds: lower 95, upper 105
    Track object 2 state Down decrement 5
  Load balancing: host-dependent
  There is 1 forwarder (1 active)
  Forwarder 1
    State is Active
      1 state change, last state change 23:50:15
    MAC address is 0007.b400.0101 (default)
    Owner ID is 0005.0050.6c08
    Redirection enabled
    Preemption enabled, min delay 60 sec
    Active is local, weighting 105
```

Customizing GLBP

Customizing the behavior of GLBP is optional. Be aware that as soon as you enable a GLBP group, that group is operating. It is possible that if you first enable a GLBP group before customizing GLBP, the device could take over control of the group and become the AVG before you have finished customizing the feature. Therefore, if you plan to customize GLBP, it is a good idea to do so before enabling GLBP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]

5. **glbp** *group* **timers** [**msec**] *hellotime* [**msec**] *holdtime*
6. **glbp** *group* **timers redirect** *redirect timeout*
7. **glbp** *group* **load-balancing** [**host-dependent** | **round-robin** | **weighted**]
8. **glbp** *group* **priority** *level*
9. **glbp** *group* **preempt** [**delay minimum** *seconds*]
10. **glbp** *group* **name** *redundancy-name*
11. **exit**
12. **no glbp sso**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/0`	Specifies an interface type and number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.21.8.32 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**glbp** *group* **timers** [**msec**] *hellotime* [**msec**] *holdtime* **Example:** `Device(config-if)# glbp 10 timers 5 18`	Configures the interval between successive hello packets sent by the AVG in a GLBP group. • The *holdtime* argument specifies the interval in seconds before the virtual gateway and virtual forwarder information in the hello packet is considered invalid. • The optional **msec** keyword specifies that the following argument will be expressed in milliseconds, instead of the default seconds.

	Command or Action	Purpose
Step 6	**glbp** *group* **timers redirect** *redirect timeout* **Example:** `Device(config-if)# glbp 10 timers redirect 1800 28800`	Configures the time interval during which the AVG continues to redirect clients to an AVF. The default is 600 seconds (10 minutes). • The *timeout* argument specifies the interval in seconds before a secondary virtual forwarder becomes invalid. The default is 14,400 seconds (4 hours). **Note** The zero value for the *redirect* argument cannot be removed from the range of acceptable values because preexisting configurations of Cisco IOS software already using the zero value could be negatively affected during an upgrade. However, a zero setting is not recommended and, if used, results in a redirect timer that never expires. If the redirect timer does not expire, and the device fails, new hosts continue to be assigned to the failed device instead of being redirected to the backup.
Step 7	**glbp** *group* **load-balancing** [**host-dependent** | **round-robin** | **weighted**] **Example:** `Device(config-if)# glbp 10 load-balancing host-dependent`	Specifies the method of load balancing used by the GLBP AVG.
Step 8	**glbp** *group* **priority** *level* **Example:** `Device(config-if)# glbp 10 priority 254`	Sets the priority level of the gateway within a GLBP group. • The default value is 100.
Step 9	**glbp** *group* **preempt** [**delay minimum** *seconds*] **Example:** `Device(config-if)# glbp 10 preempt delay minimum 60`	Configures the device to take over as AVG for a GLBP group if it has a higher priority than the current AVG. • This command is disabled by default. • Use the optional **delay** and **minimum** keywords and the *seconds* argument to specify a minimum delay interval in seconds before preemption of the AVG takes place.
Step 10	**glbp** *group* **name** *redundancy-name* **Example:** `Device(config-if)# glbp 10 name abc123`	Enables IP redundancy by assigning a name to the GLBP group. • The GLBP redundancy client must be configured with the same GLBP group name so the redundancy client and the GLBP group can be connected.
Step 11	**exit** **Example:**	Exits interface configuration mode, and returns the device to global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# exit`	
Step 12	**no glbp sso** **Example:** `Device(config)# no glbp sso`	(Optional) Disables GLBP support of SSO.

Configuring GLBP MD5 Authentication Using a Key String

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **glbp** *group-number* **authentication md5 key-string** [**0** | **7**] *key*
6. **glbp** *group-number* **ip** [*ip-address* [**secondary**]]
7. Repeat Steps 1 through 6 on each device that will communicate.
8. **end**
9. **show glbp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.

	Command or Action	Purpose
Step 5	**glbp** *group-number* **authentication md5 key-string** [**0** | **7**] *key* **Example:** `Device(config-if)# glbp 1 authentication md5 key-string d00b4r987654321a`	Configures an authentication key for GLBP MD5 authentication. • The key string cannot exceed 100 characters in length. • No prefix to the *key* argument or specifying **0** means the key is unencrypted. • Specifying **7** means the key is encrypted. The key-string authentication key will automatically be encrypted if the **service password-encryption** global configuration command is enabled.
Step 6	**glbp** *group-number* **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# glbp 1 ip 10.0.0.10`	Enables GLBP on an interface and identifies the primary IP address of the virtual gateway.
Step 7	Repeat Steps 1 through 6 on each device that will communicate.	—
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 9	**show glbp** **Example:** `Device# show glbp`	(Optional) Displays GLBP information. • Use this command to verify your configuration. The key string and authentication type will be displayed if configured.

Configuring GLBP MD5 Authentication Using a Key Chain

Perform this task to configure GLBP MD5 authentication using a key chain. Key chains allow a different key string to be used at different times according to the key chain configuration. GLBP will query the appropriate key chain to obtain the current live key and key ID for the specified key chain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*
5. **key-string** *string*
6. **exit**
7. **exit**
8. **interface** *type number*

9. **ip address** *ip-address mask* [**secondary**]
10. **glbp** *group-number* **authentication md5 key-chain** *name-of-chain*
11. **glbp** *group-number* **ip** [*ip-address* [**secondary**]]
12. Repeat Steps 1 through 10 on each device that will communicate.
13. **end**
14. **show glbp**
15. **show key chain**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain glbp2`	Enables authentication for routing protocols and identifies a group of authentication keys and enters key-chain configuration mode.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 100`	Identifies an authentication key on a key chain. • The value for the *key-id* argument must be a number.
Step 5	**key-string** *string* **Example:** `Device(config-keychain-key)# key-string abc123`	Specifies the authentication string for a key and enters key-chain key configuration mode. • The value for the *string* argument can be 1 to 80 uppercase or lowercase alphanumeric characters; the first character cannot be a numeral.
Step 6	**exit** **Example:** `Device(config-keychain-key)# exit`	Returns to key-chain configuration mode.
Step 7	**exit** **Example:** `Device(config-keychain)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 8	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/1`	Configures an interface type and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.21.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 10	**glbp** *group-number* **authentication md5 key-chain** *name-of-chain* **Example:** `Device(config-if)# glbp 1 authentication md5 key-chain glbp2`	Configures an authentication MD5 key chain for GLBP MD5 authentication. • The key chain name must match the name specified in Step 3.
Step 11	**glbp** *group-number* **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# glbp 1 ip 10.21.0.12`	Enables GLBP on an interface and identifies the primary IP address of the virtual gateway.
Step 12	Repeat Steps 1 through 10 on each device that will communicate.	—
Step 13	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 14	**show glbp** **Example:** `Device# show glbp`	(Optional) Displays GLBP information. • Use this command to verify your configuration. The key chain and authentication type will be displayed if configured.
Step 15	**show key chain** **Example:** `Device# show key chain`	(Optional) Displays authentication key information.

Configuring GLBP Text Authentication

Text authentication provides minimal security. Use MD5 authentication if security is required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **glbp** *group-number* **authentication text** *string*
6. **glbp** *group-number* **ip** [*ip-address* [**secondary**]]
7. Repeat Steps 1 through 6 on each device that will communicate.
8. **end**
9. **show glbp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**glbp** *group-number* **authentication text** *string* **Example:** `Device(config-if)# glbp 10 authentication text stringxyz`	Authenticates GLBP packets received from other devices in the group. • If you configure authentication, all devices within the GLBP group must use the same authentication string.
Step 6	**glbp** *group-number* **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# glbp 1 ip 10.0.0.10`	Enables GLBP on an interface and identifies the primary IP address of the virtual gateway.
Step 7	Repeat Steps 1 through 6 on each device that will communicate.	—

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 9	**show glbp** **Example:** `Device# show glbp`	(Optional) Displays GLBP information. • Use this command to verify your configuration.

Configuring GLBP Weighting Values and Object Tracking

GLBP weighting is used to determine whether a GLBP group can act as a virtual forwarder. Initial weighting values can be set and optional thresholds specified. Interface states can be tracked and a decrement value set to reduce the weighting value if the interface goes down. When the GLBP group weighting drops below a specified value, the group will no longer be an active virtual forwarder. When the weighting rises above a specified value, the group can resume its role as an active virtual forwarder.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **interface** *type number* {**line-protocol** | **ip routing**}
4. **exit**
5. **interface** *type number*
6. **glbp** *group* **weighting** *maximum* [**lower** *lower*] [**upper** *upper*]
7. **glbp** *group* **weighting track** *object-number* [**decrement** *value*]
8. **glbp** *group* **forwarder preempt** [**delay minimum** *seconds*]
9. **exit**
10. **show track** [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**track** *object-number* **interface** *type number* {**line-protocol** | **ip routing**} **Example:** `Device(config)# track 2 interface POS 6/0/0 ip routing`	Configures an interface to be tracked where changes in the state of the interface affect the weighting of a GLBP gateway, and enters tracking configuration mode. • This command configures the interface and corresponding object number to be used with the **glbp weighting track** command. • The **line-protocol** keyword tracks whether the interface is up. The **ip routing** keywords also check that IP routing is enabled on the interface, and an IP address is configured.
Step 4	**exit** **Example:** `Device(config-track)# exit`	Returns to global configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Enters interface configuration mode.
Step 6	**glbp** *group* **weighting** *maximum* [**lower** *lower*] [**upper** *upper*] **Example:** `Device(config-if)# glbp 10 weighting 110 lower 95 upper 105`	Specifies the initial weighting value, and the upper and lower thresholds, for a GLBP gateway.
Step 7	**glbp** *group* **weighting track** *object-number* [**decrement** *value*] **Example:** `Device(config-if)# glbp 10 weighting track 2 decrement 5`	Specifies an object to be tracked that affects the weighting of a GLBP gateway. • The *value* argument specifies a reduction in the weighting of a GLBP gateway when a tracked object fails.
Step 8	**glbp** *group* **forwarder preempt** [**delay minimum** *seconds*] **Example:** `Device(config-if)# glbp 10 forwarder preempt delay minimum 60`	Configures the device to take over as AVF for a GLBP group if the current AVF for a GLBP group falls below its low weighting threshold. • This command is enabled by default with a delay of 30 seconds. • Use the optional **delay** and **minimum** keywords and the *seconds* argument to specify a minimum delay interval in seconds before preemption of the AVF takes place.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Device(config-if)# exit`	Returns to privileged EXEC mode.
Step 10	**show track** [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**] **Example:** `Device# show track 2`	Displays tracking information.

Troubleshooting GLBP

GLBP introduces five privileged EXEC mode commands to enable display of diagnostic output concerning various events relating to the operation of GLBP. The **debug condition glbp**,**debug glbp errors**, **debug glbp events**, **debug glbp packets**, and **debug glbp terse** commands are intended only for troubleshooting purposes because the volume of output generated by the software can result in severe performance degradation on the device. Perform this task to minimize the impact of using the **debug glbp** commands.

This procedure will minimize the load on the device created by the **debug condition glbp**or **debug glbp** command because the console port is no longer generating character-by-character processor interrupts. If you cannot connect to a console directly, you can run this procedure via a terminal server. If you must break the Telnet connection, however, you may not be able to reconnect because the device may be unable to respond due to the processor load of generating the debugging output.

Before you begin

This task requires a device running GLBP to be attached directly to a console.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no logging console**
4. Use Telnet to access a device port and repeat Steps 1 and 2.
5. **end**
6. **terminal monitor**
7. **debug condition glbp** *interface-type interface-number group* [*forwarder*]
8. **terminal no monitor**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no logging console** **Example:** `Device(config)# no logging console`	Disables all logging to the console terminal. • To reenable logging to the console, use the**logging console** command in global configuration mode.
Step 4	Use Telnet to access a device port and repeat Steps 1 and 2.	Enters global configuration mode in a recursive Telnet session, which allows the output to be redirected away from the console port.
Step 5	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.
Step 6	**terminal monitor** **Example:** `Device# terminal monitor`	Enables logging output on the virtual terminal.
Step 7	**debug condition glbp** *interface-type interface-number group* [*forwarder*] **Example:** `Device# debug condition glbp GigabitEthernet0/0/0 1`	Displays debugging messages about GLBP conditions. • Try to enter only specific **debug condition glbp** or **debug glbp** commands to isolate the output to a certain subcomponent and minimize the load on the processor. Use appropriate arguments and keywords to generate more detailed debug information on specified subcomponents. • Enter the specific **no debug condition glbp** or **no debug glbp** command when you are finished.
Step 8	**terminal no monitor** **Example:** `Device# terminal no monitor`	Disables logging on the virtual terminal.

Configuration Examples for GLBP

Example: Customizing GLBP Configuration

```
Device(config)# interface fastethernet 0/0
Device(config-if)# ip address 10.21.8.32 255.255.255.0
Device(config-if)# glbp 10 timers 5 18
Device(config-if)# glbp 10 timers redirect 1800 28800
Device(config-if)# glbp 10 load-balancing host-dependent
Device(config-if)# glbp 10 priority 254
Device(config-if)# glbp 10 preempt delay minimum 60

Device(config-if)# glbp 10 client-cache maximum 1200 timeout 245
```

Example: Configuring GLBP MD5 Authentication Using Key Strings

The following example shows how to configure GLBP MD5 authentication using a key string:

```
Device(config)# interface Ethernet 0/1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# glbp 2 authentication md5 key-string ThisStringIsTheSecretKey
Device(config-if)# glbp 2 ip 10.0.0.10
```

Example: Configuring GLBP MD5 Authentication Using Key Chains

In the following example, GLBP queries the key chain “AuthenticateGLBP” to obtain the current live key and key ID for the specified key chain:

```
Device(config)# key chain AuthenticateGLBP
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string ThisIsASecretKey
Device(config-keychain-key)# exit
Device(config-keychain)# exit
Device(config)# interface Ethernet 0/1
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# glbp 2 authentication md5 key-chain AuthenticateGLBP
Device(config-if)# glbp 2 ip 10.0.0.10
```

Example: Configuring GLBP Text Authentication

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.21.8.32 255.255.255.0
Device(config-if)# glbp 10 authentication text stringxyz
Device(config-if)# glbp 10 ip 10.21.8.10
```

Example: Configuring GLBP Weighting

In the following example, the device is configured to track the IP routing state of the POS interface 5/0/0 and 6/0/0, an initial GLBP weighting with upper and lower thresholds is set, and a weighting decrement value of 10 is set. If POS interface 5/0/0 and 6/0/0 go down, the weighting value of the device is reduced.

```
Device(config)# track 1 interface POS 5/0/0 ip routing
Device(config)# track 2 interface POS 6/0/0 ip routing
Device(config)# interface fastethernet 0/0/0
Device(config-if)# glbp 10 weighting 110 lower 95 upper 105
Device(config-if)# glbp 10 weighting track 1 decrement 10
Device(config-if)# glbp 10 weighting track 2 decrement 10
Device(config-if)# glbp 10 forwarder preempt delay minimum 60
```

Example: Enabling GLBP Configuration

In the following example, the device is configured to enable GLBP, and the virtual IP address of 10.21.8.10 is specified for GLBP group 10:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.21.8.32 255.255.255.0
Device(config-if)# glbp 10 ip 10.21.8.10
```

Additional References for GLBP

Related Documents

Related Topic	Document Title
GLBP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Application Services Command Reference
In Service Software Upgrade (ISSU) configuration	"In Service Software Upgrade" process module in the *Cisco IOS High Availability Configuration Guide*
Key chains and key management commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing Protocol-Independent Command Reference*
Object tracking	"Configuring Enhanced Object Tracking" module
Stateful Switchover	The "Stateful Switchover" module in the *Cisco IOS High Availability Configuration Guide*
VRRP	"Configuring VRRP" module
HSRP	"Configuring HSRP" module
GLBP Support for IPv6	“FHRP - GLBP Support for IPv6” module

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for GLBP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Glossary

active RP—The Route Processor (RP) controls the system, provides network services, runs routing protocols and presents the system management interface.

AVF—active virtual forwarder. One virtual forwarder within a GLBP group is elected as active virtual forwarder for a specified virtual MAC address, and it is responsible for forwarding packets sent to that MAC address. Multiple active virtual forwarders can exist for each GLBP group.

AVG—active virtual gateway. One virtual gateway within a GLBP group is elected as the active virtual gateway, and is responsible for the operation of the protocol.

GLBP gateway—Gateway Load Balancing Protocol gateway. A router or gateway running GLBP. Each GLBP gateway may participate in one or more GLBP groups.

GLBP group—Gateway Load Balancing Protocol group. One or more GLBP gateways configured with the same GLBP group number on connected Ethernet interfaces.

ISSU—In Service Software Upgrade. A process that allows Cisco IOS XE software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco IOS XE software to be modified while packet forwarding continues, which increases network availability and reduces downtime caused by planned software upgrades.

NSF—nonstop forwarding. The ability of a router to continue to forward traffic to a router that may be recovering from a failure. Also, the ability of a router recovering from a failure to continue to correctly forward traffic sent to it by a peer.

RP—Route Processor. A generic term for the centralized control unit in a chassis. Platforms usually use a platform-specific term, such as RSP on the Cisco 7500, the PRE on the Cisco 10000, or the SUP+MSFC on the Cisco 7600.

RPR—Route Processor Redundancy. RPR provides an alternative to the High System Availability (HSA) feature. HSA enables a system to reset and use a standby Route Processor (RP) if the active RP fails. Using RPR, you can reduce unplanned downtime because RPR enables a quicker switchover between an active and standby RP if the active RP experiences a fatal error.

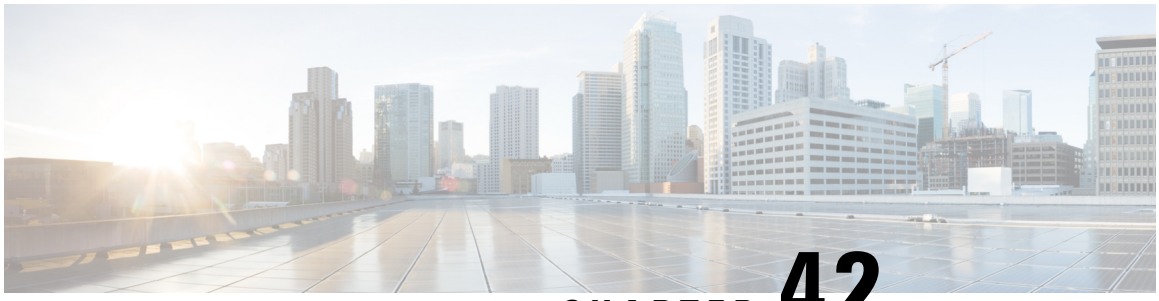
RPR+—An enhancement to RPR in which the standby RP is fully initialized.

SSO—Stateful Switchover. Enables applications and features to maintain state information between an active and standby unit.

standby RP—An RP that has been fully initialized and is ready to assume control from the active RP should a manual or fault-induced switchover occur.

switchover—An event in which system control and routing protocol execution are transferred from the active RP to the standby RP. Switchover may be a manual operation or may be induced by a hardware or software fault. Switchover may include transfer of the packet forwarding function in systems that combine system control and packet forwarding in an indivisible unit.

vIP—virtual IP address. An IPv4 address. There must be only one virtual IP address for each configured GLBP group. The virtual IP address must be configured on at least one GLBP group member. Other GLBP group members can learn the virtual IP address from hello messages.



CHAPTER 42

HSRP for IPv6

IPv6 routing protocols ensure device-to-device resilience and failover. However, in situations in which the path between a host and the first-hop device fails, or the first-hop device itself fails, first hop redundancy protocols (FHRPs) ensure host-to-device resilience and failover.

The Hot Standby Router Protocol (HSRP) protects data traffic in case of a gateway failure.

Prerequisites for HSRP for IPv6

HSRP version 2 must be enabled on an interface before HSRP for IPv6 can be configured.

Information About HSRP for IPv6

HSRP for IPv6 Overview

The HSRP is an FHRP designed to allow for transparent failover of the first-hop IP device. HSRP provides high network availability by providing first-hop routing redundancy for IP hosts on Ethernet configured with a default gateway IP address. HSRP is used in a group of devices for selecting an active device and a standby device. In a group of device interfaces, the active device is the device of choice for routing packets; the standby device is the device that takes over when the active device fails or when preset conditions are met.

IPv6 hosts learn of available IPv6 devices through IPv6 neighbor discovery RA messages. These are multicast periodically, or may be solicited by hosts. HSRP is designed to provide only a virtual first hop for IPv6 hosts.

An HSRP IPv6 group has a virtual MAC address that is derived from the HSRP group number, and a virtual IPv6 link-local address that is, by default, derived from the HSRP virtual MAC address. Periodic RAs are sent for the HSRP virtual IPv6 link-local address when the HSRP group is active. These RAs stop after a final RA is sent when the group leaves the active state.

Periodic RAs for the interface link-local address stop after a final RA is sent while at least one virtual IPv6 link-local address is configured on the interface. No restrictions occur for the interface IPv6 link-local address other than that mentioned for the RAs. Other protocols continue to receive and send packets to this address.

HSRP uses a priority mechanism to determine which HSRP configured device is to be the default active device. To configure a device as the active device, you assign it a priority that is higher than the priority of all the other HSRP-configured devices. The default priority is 100, so if you configure just one device to have a higher priority, that device will be the default active device.

HSRP IPv6 Virtual MAC Address Range

HSRP IPv6 uses a different virtual MAC address block than does HSRP for IP:

0005.73A0.0000 through 0005.73A0.0FFF (4096 addresses)

HSRP IPv6 UDP Port Number

Port number 2029 has been assigned to HSRP IPv6.

How to Enable HSRP for IPv6

Enabling an HSRP Group for IPv6 Operation

HSRP version 2 must be enabled on an interface before HSRP IPv6 can be configured.

Enabling HSRP Version 2

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **standby version** {**1** | **2**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 4	**standby version** {**1** | **2**} **Example:** `Device(config-if)# standby version 2`	Changes the version of the HSRP. • Version 1 is the default.

Enabling and Verifying an HSRP Group for IPv6 Operation

In this task, when you enter the **standby ipv6** command, a link-local address is generated from the link-local prefix, and a modified EUI-64 format interface identifier is generated in which the EUI-64 interface identifier is created from the relevant HSRP virtual MAC address.

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 (1111 1110 10) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate; the nodes do not need site-local or globally unique addresses to communicate.

In IPv6, a device on the link advertises in RA messages any site-local and global prefixes, and its willingness to function as a default device for the link. RA messages are sent periodically and in response to router solicitation messages, which are sent by hosts at system startup.

A node on the link can automatically configure site-local and global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the RA messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the RA messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **standby** [*group-number*] **ipv6** {*link-local-address* | **autoconfig**}
6. **standby** [*group-number*] **preempt** [**delay minimum** *seconds* | **reload** *seconds* | **sync** *seconds*]
7. **standby** [*group-number*] **priority** *priority*
8. **exit**
9. **show standby** [*type number* [*group*]] [**all** | **brief**]
10. **show ipv6 interface** [**brief**] [*interface-type interface-number*] [**prefix**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams. • The **ipv6 unicast-routing** command must be enabled for HSRP for IPv6 to work.
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 5	**standby** [*group-number*] **ipv6** {*link-local-address* | **autoconfig**} **Example:** `Device(config-if)# standby 1 ipv6 autoconfig`	Activates the HSRP in IPv6.
Step 6	**standby** [*group-number*] **preempt** [**delay minimum** *seconds* | **reload** *seconds* | **sync** *seconds*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption and preemption delay.
Step 7	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Returns the device to privileged EXEC mode.
Step 9	**show standby** [*type number* [*group*]] [**all** | **brief**] **Example:** `Device# show standby`	Displays HSRP information.

	Command or Action	Purpose
Step 10	**show ipv6 interface** [**brief**] [*interface-type interface-number*] [**prefix**] **Example:** `Device# show ipv6 interface GigabitEthernet 0/0/0`	Displays the usability status of interfaces configured for IPv6.

Configuration Examples for HSRP for IPv6

Example: Configuration and Verification for an HSRP Group

The following example shows configuration and verification for an HSRP group for IPv6 that consists of Device1 and Device2. The **show standby** command is issued for each device to verify the device's configuration:

Device 1 configuration

```
interface FastEthernet0/0.100
description DATA VLAN for PCs
encapsulation dot1Q 100
ipv6 address 2001:DB8:CAFE:2100::BAD1:1010/64
standby version 2
standby 101 priority 120
standby 101 preempt delay minimum 30
standby 101 authentication ese
standby 101 track Serial0/1/0.17 90
standby 201 ipv6 autoconfig
standby 201 priority 120
standby 201 preempt delay minimum 30
standby 201 authentication ese
standby 201 track Serial0/1/0.17 90
Device1# show standby
FastEthernet0/0.100 - Group 101 (version 2)
State is Active
2 state changes, last state change 5w5d
Active virtual MAC address is 0000.0c9f.f065
Local virtual MAC address is 0000.0c9f.f065 (v2 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 2.296 secs
Authentication text "ese"
Preemption enabled, delay min 30 secs
Active router is local
Priority 120 (configured 120)
Track interface Serial0/1/0.17 state Up decrement 90
IP redundancy name is "hsrp-Fa0/0.100-101" (default)
FastEthernet0/0.100 - Group 201 (version 2)
State is Active
2 state changes, last state change 5w5d
Virtual IP address is FE80::5:73FF:FEA0:C9
Active virtual MAC address is 0005.73a0.00c9
Local virtual MAC address is 0005.73a0.00c9 (v2 IPv6 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 2.428 secs
Authentication text "ese"
Preemption enabled, delay min 30 secs
```

```
Active router is local
Standby router is FE80::20F:8FFF:FE37:3B70, priority 100 (expires in 7.856 sec)
Priority 120 (configured 120)
Track interface Serial0/1/0.17 state Up decrement 90
IP redundancy name is "hsrp-Fa0/0.100-201" (default)
```

Device 2 configuration

```
interface FastEthernet0/0.100
description DATA VLAN for Computers
encapsulation dot1Q 100
ipv6 address 2001:DB8:CAFE:2100::BAD1:1020/64
standby version 2
standby 101 preempt
standby 101 authentication ese
standby 201 ipv6 autoconfig
standby 201 preempt
standby 201 authentication ese
Device2# show standby
FastEthernet0/0.100 - Group 101 (version 2)
State is Standby
7 state changes, last state change 5w5d
Active virtual MAC address is 0000.0c9f.f065
Local virtual MAC address is 0000.0c9f.f065 (v2 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.936 secs
Authentication text "ese"
Preemption enabled
MAC address is 0012.7fc6.8f0c
Standby router is local
Priority 100 (default 100)
IP redundancy name is "hsrp-Fa0/0.100-101" (default)
FastEthernet0/0.100 - Group 201 (version 2)
State is Standby
7 state changes, last state change 5w5d
Virtual IP address is FE80::5:73FF:FEA0:C9
Active virtual MAC address is 0005.73a0.00c9
Local virtual MAC address is 0005.73a0.00c9 (v2 IPv6 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.936 secs
Authentication text "ese"
Preemption enabled
Active router is FE80::212:7FFF:FEC6:8F0C, priority 120 (expires in 7.548 sec)
MAC address is 0012.7fc6.8f0c
Standby router is local
Priority 100 (default 100)
IP redundancy name is "hsrp-Fa0/0.100-201" (default)
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
VRRP commands	*Cisco IOS IP Application Services Command Reference*
Object tracking	Configuring Enhanced Object Tracking

Related Topic	Document Title
Hot Standby Routing Protocol (HSRP)	Configuring HSRP
In Service Software Upgrace (ISSU)	"In Service Software Upgrade Process" in the *High Availability Configuration Guide*
Gateway Load Balancing Protocol (GLBP)	Configuring GLBP
Stateful Switchover	The Stateful Switchover section in the*High Availability Configuration Guide*

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
VRRP MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2338	Virtual Router Redundancy Protocol
RFC 2787	Definitions of Managed Objects for the Virtual Router Redundancy Protocol
RFC 3768	Virtual Router Redundancy Protocol (VRRP)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HSRP for IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Glossary

- **CPE** --Customer premises equipment
- **FHRP** --First hop redundancy protocol
- **GLBP** --Gateway load balancing protocol
- **HSRP** --Hot standby routing protocol
- **NA** --Neighbor advertisement
- **ND** --Neighbor Discovery
- **NS** --Neighbor solicitation
- **PE** --Provider equipment
- **RA** --Router advertisement
- **RS** --Router solicitation

CHAPTER 43

Configuring HSRP

The Hot Standby Router Protocol (HSRP) is a First Hop Redundancy Protocol (FHRP) designed to allow for transparent failover of the first-hop IP device. HSRP provides high network availability by providing first-hop routing redundancy for IP hosts on networks configured with a default gateway IP address. HSRP is used in a group of routers for selecting an active device and a standby device. In a group of device interfaces, the active device is the device of choice for routing packets; the standby device is the device that takes over when the active device fails or when preset conditions are met.

Restrictions for HSRP

- HSRP is designed for use over multiaccess, multicast, or broadcast capable Ethernet LANs. HSRP is not intended as a replacement for existing dynamic protocols.

Information About HSRP

HSRP Operation

Most IP hosts have an IP address of a single device configured as the default gateway. When HSRP is used, the HSRP virtual IP address is configured as the host's default gateway instead of the IP address of the device.

HSRP is useful for hosts that do not support a discovery protocol (such as ICMP Router Discovery Protocol [IRDP]) and cannot switch to a new device when their selected device reloads or loses power. Because existing TCP sessions can survive the failover, this protocol also provides a more transparent recovery for hosts that dynamically choose a next hop for routing IP traffic.

When HSRP is configured on a network segment, it provides a virtual MAC address and an IP address that is shared among a group of devices running HSRP. The address of this HSRP group is referred to as the *virtual*

IP address. One of these devices is selected by the protocol to be the active device. The active device receives and routes packets destined for the MAC address of the group. For *n* devices running HSRP, *n*+ 1 IP and MAC addresses are assigned.

HSRP detects when the designated active device fails, at which point a selected standby device assumes control of the MAC and IP addresses of the Hot Standby group. A new standby device is also selected at that time.

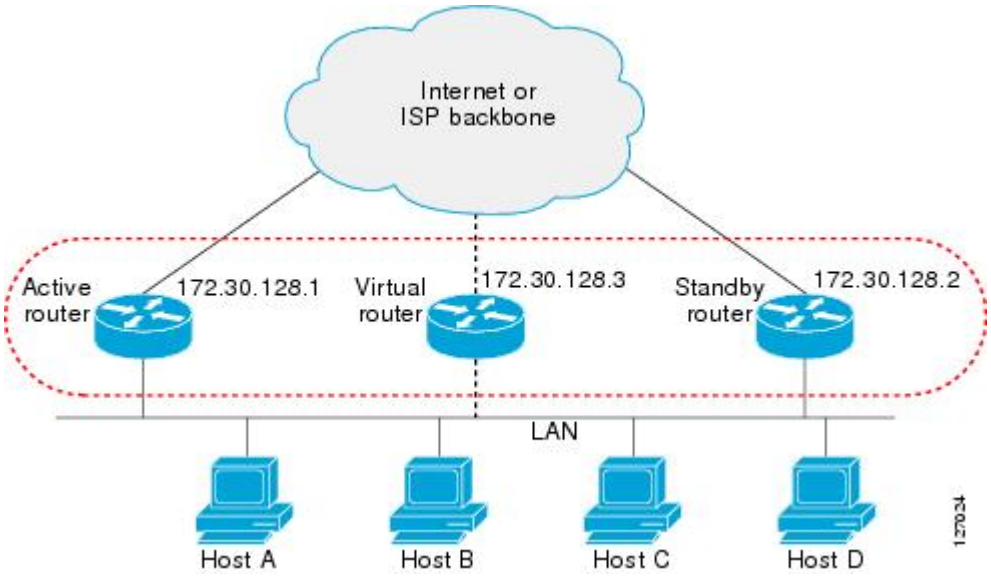
HSRP uses a priority mechanism to determine which HSRP configured device is to be the default active device. To configure a device as the active device, you assign it a priority that is higher than the priority of all the other HSRP-configured devices. The default priority is 100, so if you configure just one device to have a higher priority, that device will be the default active device.

Devices that are running HSRP send and receive multicast UDP-based hello messages to detect device failure and to designate active and standby devices. When the active device fails to send a hello message within a configurable period of time, the standby device with the highest priority becomes the active device. The transition of packet forwarding functions between devices is completely transparent to all hosts on the network.

You can configure multiple Hot Standby groups on an interface, thereby making fuller use of redundant devices and load sharing.

The figure below shows a network configured for HSRP. By sharing a virtual MAC address and IP address, two or more devices can act as a single *virtual router*. The virtual device does not physically exist but represents the common default gateway for devices that are configured to provide backup to each other. You do not need to configure the hosts on the LAN with the IP address of the active device. Instead, you configure them with the IP address (virtual IP address) of the virtual device as their default gateway. If the active device fails to send a hello message within the configurable period of time, the standby device takes over and responds to the virtual addresses and becomes the active device, assuming the active device duties.

Figure 10: HSRP Topology



HSRP Version 2 Design

HSRP version 2 is designed to address the following restrictions in HSRP version 1:

- In HSRP version 1, millisecond timer values are not advertised or learned. HSRP version 2 advertises and learns millisecond timer values. This change ensures stability of the HSRP groups in all cases.
- In HSRP version 1, group numbers are restricted to the range from 0 to 255. HSRP version 2 expands the group number range from 0 to 4095.
- HSRP version 2 provides improved management and troubleshooting. With HSRP version 1, you cannot use HSRP active hello messages to identify which physical device sent the message because the source MAC address is the HSRP virtual MAC address. The HSRP version 2 packet format includes a 6-byte identifier field that is used to uniquely identify the sender of the message. Typically, this field is populated with the interface MAC address.
- The multicast address 224.0.0.2 is used to send HSRP hello messages. This address can conflict with Cisco Group Management Protocol (CGMP) leave processing.

Version 1 is the default version of HSRP.

HSRP version 2 uses the new IP multicast address 224.0.0.102 to send hello packets instead of the multicast address of 224.0.0.2, used by HSRP version 1. This new multicast address allows CGMP leave processing to be enabled at the same time as HSRP.

HSRP version 2 permits an expanded group number range, 0 to 4095, and consequently uses a new MAC address range 0000.0C9F.F000 to 0000.0C9F.FFFF. The increased group number range does not imply that an interface can, or should, support that many HSRP groups. The expanded group number range was changed to allow the group number to match the VLAN number on subinterfaces.

When the HSRP version is changed, each group will reinitialize because it now has a new virtual MAC address.

HSRP version 2 has a different packet format than HSRP version 1. The packet format uses a type-length-value (TLV) format. HSRP version 2 packets received by an HSRP version 1 device will have the type field mapped to the version field by HSRP version 1 and subsequently ignored.

The Gateway Load Balancing Protocol (GLBP) also addresses the same restrictions relative to HSRP version 1 that HSRP version 2 does. See the *Configuring GLBP* document for more information on GLBP.

Jitter timers

Jitter timers are used in HSRP. They are recommended for timers running on services that work realtime and scale. Jitter timers are intended to significantly improve the reliability of HSRP, and other FHRP protocols, by reducing the chance of bunching of HSRP groups operations, and thus help reduce CPU and network traffic spikes. In the case of HSRP, a given device may have up to 4000 operational groups configured. In order to distribute the load on the device and network, the HSRP timers use a jitter. A given timer instance may take up to 20% more than the configured value. For example, for a hold time set to 15 seconds, the actual hold time may take 18 seconds.

In HSRP, the Hello timer (which sends the Hello Packet) has a negative Jitter, while the Holddown timer (which checks for failure of a peer) has a positive jitter.

HSRP Configuration Changes

With CSCsv12265, an HSRP group may be configured with a virtual IP address that matches the subnet of an IP address of a secondary interface.

When the virtual IP address of an HSRP group is configured with the same network ID as a secondary interface IP address, the source address of HSRP messages is automatically set to the most appropriate interface address. This configuration change allows the following configuration:

```
interface Ethernet1/0
 ip address 192.168.1.1 255.255.255.0
 ip address 192.168.2.1 255.255.255.0 secondary
 standby 1 ip 192.168.1.254
 standby 1 priority 105
 standby 1 preempt
 standby 2 ip 192.168.2.254 !Same network ID as secondary interface
```

Prior to CSCsv12265, an HSRP group remained in INIT state unless the HSRP virtual IP address had the same network ID as the primary interface address.

In addition, the following warning message is displayed if an HSRP group address is configured when no interface addresses are configured:

```
% Warning: address is not within a subnet on this interface
```

HSRP Benefits

Redundancy

HSRP employs a redundancy scheme that is time proven and deployed extensively in large networks.

Fast Failover

HSRP provides transparent fast failover of the first-hop device.

Preemption

Preemption allows a standby device to delay becoming active for a configurable amount of time.

Authentication

HSRP message digest 5 (MD5) algorithm authentication protects against HSRP-spoofing software and uses the industry-standard MD5 algorithm for improved reliability and security.

HSRP Groups and Group Attributes

You can use the CLI to apply group attributes to:

- A single HSRP group—performed in interface configuration mode and applies to a group.
- All groups on the interface—performed in interface configuration mode and applies to all groups on the interface.
- All groups on all interfaces—performed in global configuration mode and applies to all groups on all interfaces.

HSRP Preemption

When a newly reloaded device becomes HSRP active, and there is already an HSRP active device on the network, HSRP preemption may appear to not function. HSRP preemption may appear not function correctly

because the new HSRP active device did not receive any hello packets from the current HSRP active device, and the preemption configuration never factored into the new device's decision making.

HSRP may appear to not function on some larger hardware platforms where there can be a delay in an interface receiving packets.

In general, we recommend that all HSRP devices have the following configuration:

standby delay minimum 30 reload 60

The **standby delay minimum reload** interface configuration command delays HSRP groups from initializing for the specified time after the interface comes up.

This is a different command than the **standby preempt delay** interface configuration command, which enables HSRP preemption delay.

HSRP Priority and Preemption

Preemption enables the HSRP router with the highest priority to immediately become the active router. Priority is determined first by the configured priority value, and then by the IP address. In case of ties, the primary IP addresses are compared, and the higher IP address has priority. This scenario only occurs during the initial election phase when no Active router is currently present. Once an Active router has been elected, a router with a higher IP address will not cause a preemption, in order to avoid unnecessary disruptions. The only way to preempt the current Active router is by increasing its priority.

In each case, a higher value is of greater priority. If you do not use the **standby preempt** interface configuration command in the configuration for a router, that router will not become the active router, even if its priority is higher than all other routers.

A standby router with equal priority but a higher IP address will not preempt the active router.

When a router first comes up, it does not have a complete routing table. You can set a preemption delay that allows preemption to be delayed for a configurable time period. This delay period allows the router to populate its routing table before becoming the active router.

If preemption is not enabled, then a router may appear to preempt the active router if it does not receive any Hello messages from the active router.

How Object Tracking Affects the Priority of an HSRP Device

The priority of a device can change dynamically if it has been configured for object tracking and the object that is being tracked goes down. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to HSRP, either immediately or after a specified delay. The object values are reported as either up or down. Examples of objects that can be tracked are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, the HSRP priority is reduced. The HSRP device with the higher priority can become the active device if it has the **standby preempt** command configured.

HSRP Addressing

HSRP devices communicate between each other by exchanging HSRP hello packets. These packets are sent to the destination IP multicast address 224.0.0.2 (reserved multicast address used to communicate to all devices) on UDP port 1985. The active device sources hello packets from its configured IP address and the

HSRP virtual MAC address while the standby device sources hellos from its configured IP address and the interface MAC address, which may or may not be the burned-in MAC address (BIA).

Because hosts are configured with their default gateway as the HSRP virtual IP address, hosts must communicate with the MAC address associated with the HSRP virtual IP address. This MAC address will be a virtual MAC address in the format of 0000.0C07.AC*xy*, where *xy* is the HSRP group number in hexadecimal based on the respective interface. For example, HSRP group one will use the HSRP virtual MAC address of 0000.0C07.AC01. Hosts on the adjoining LAN segment use the normal Address Resolution Protocol (ARP) process to resolve the associated MAC addresses.

HSRP version 2 uses the new IP multicast address 224.0.0.102 to send hello packets instead of the multicast address of 224.0.0.2, which is used by version 1. This new multicast address allows Cisco Group Management Protocol (CGMP) leave processing to be enabled at the same time as HSRP.

HSRP version 2 permits an expanded group number range, 0 to 4095, and consequently uses a new MAC address range 0000.0C9F.F000 to 0000.0C9F.FFFF.

HSRP Virtual MAC Addresses and BIA MAC Addresses

A device automatically generates a virtual MAC address for each HSRP device. However, some network implementations, such as Advanced Peer-to-Peer Networking (APPN), use the MAC address to identify the first hop for routing purposes. In this case, specify the virtual MAC address by using the **standby mac-address** command in the group; the virtual IP address is unimportant for these protocols.

The **standby use-bia** command was implemented to overcome the limitations of using a functional address for the HSRP MAC address on Token Ring interfaces. This command allows HSRP groups to use the burned-in MAC address of an interface instead of the HSRP virtual MAC address. When HSRP runs on a multiple-ring, source-routed bridging environment and the HSRP devices reside on different rings, configuring the **standby use-bia** command can prevent confusion about the routing information field (RFI).

The **standby use-bia** command is used for an interface and the **standby mac-address** command is used for an HSRP group.

HSRP Timers

For HSRP version 1, nonactive devices learn timer values from the active device, unless millisecond timer values are being used. If millisecond timer values are being used, all devices must be configured with the millisecond timer values. This rule applies if either the hello time or the hold time is specified in milliseconds. This configuration is necessary because the HSRP hello packets advertise the timer values in seconds. HSRP version 2 does not have this limitation; it advertises the timer values in milliseconds.

Jitter timers

Jitter timers are used in HSRP. They are recommended for timers running on services that work realtime and scale. Jitter timers are intended to significantly improve the reliability of HSRP, and other FHRP protocols, by reducing the chance of bunching of HSRP groups operations, and thus help reduce CPU and network traffic spikes. In the case of HSRP, a given device may have up to 4000 operational groups configured. In order to distribute the load on the device and network, the HSRP timers use a jitter. A given timer instance may take up to 20% more than the configured value. For example, for a hold time set to 15 seconds, the actual hold time may take 18 seconds.

In HSRP, the Hello timer (which sends the Hello Packet) has a negative Jitter, while the Holddown timer (which checks for failure of a peer) has a positive jitter.

HSRP MAC Refresh Interval

When HSRP runs over FDDI, you can change the interval at which a packet is sent to refresh the MAC cache on learning bridges and switches. HSRP hello packets on FDDI interfaces use the burned-in address (BIA) instead of the MAC virtual address. Refresh packets keep the MAC cache on switches and learning bridges current. Refresh packets are also used for HSRP groups configured as multigroup subordinates because these do not send regular Hello messages.

You can change the refresh interval on FDDI rings to a longer or shorter interval, thereby using bandwidth more efficiently. You can prevent the sending of any MAC refresh packets if you do not need them (if you have FDDI but do not have a learning bridge or switch).

HSRP Text Authentication

HSRP ignores unauthenticated HSRP protocol messages. The default authentication type is text authentication.

HSRP authentication protects against false HSRP hello packets causing a denial-of-service attack. For example, Device A has a priority of 120 and is the active device. If a host sends spoof HSRP hello packets with a priority of 130, then Device A stops being the active device. If Device A has authentication configured such that the spoof HSRP hello packets are ignored, Device A will remain the active device

HSRP packets will be rejected in any of the following cases:

- The authentication schemes differ on the device and in the incoming packets.
- Text authentication strings differ on the device and in the incoming packet.

HSRP MD5 Authentication

Before the introduction of HSRP MD5 authentication, HSRP authenticated protocol packets with a simple plain text string. HSRP MD5 authentication is an enhancement to generate an MD5 digest for the HSRP portion of the multicast HSRP protocol packet. This functionality provides added security and protects against the threat from HSRP-spoofing software.

MD5 authentication provides greater security than the alternative plain text authentication scheme. MD5 authentication allows each HSRP group member to use a secret key to generate a keyed MD5 hash that is part of the outgoing packet. A keyed hash of an incoming packet is generated and if the hash within the incoming packet does not match the generated hash, the packet is ignored.

The key for the MD5 hash can be either given directly in the configuration using a key string or supplied indirectly through a key chain.

HSRP has two authentication schemes:

- Plain text authentication
- MD5 authentication

HSRP authentication protects against false HSRP hello packets causing a denial-of-service attack. For example, Device A has a priority of 120 and is the active device. If a host sends spoof HSRP hello packets with a priority of 130, then Device A stops being the active device. If Device A has authentication configured such that the spoof HSRP hello packets are ignored, Device A will remain the active device.

HSRP packets will be rejected in any of the following cases:

- The authentication schemes differ on the device and in the incoming packets.
- MD5 digests differ on the device and in the incoming packet.
- Text authentication strings differ on the device and in the incoming packet.

HSRP Support for IPv6

Most IPv4 hosts have a single router's IP address configured as the default gateway. When HSRP is used, then the HSRP virtual IP address is configured as the host's default gateway instead of the router's IP address. Simple load sharing may be achieved by using two HSRP groups and configuring half the hosts with one virtual IP address and half the hosts with the other virtual IP address.

In contrast, IPv6 hosts learn of available IPv6 routers through IPv6 neighbor discovery Router Advertisement (RA) messages. These are multicast periodically, or may be solicited by hosts. HSRP is designed to provide only a virtual first hop for IPv6 hosts.

An HSRP IPv6 group has a virtual MAC address that is derived from the HSRP group number, and a virtual IPv6 link-local address that is, by default, derived from the HSRP virtual MAC address. HSRP IPv6 uses the MAC address range 0005.73A0.0000 to 0005.73A0.0FFF. Periodic RAs are sent for the HSRP virtual IPv6 link-local address when the HSRP group is active. These RAs stop after a final RA is sent when the group leaves the active state.

Periodic RAs for the interface link-local address stop after a final RA is sent while at least one virtual IPv6 link-local address is configured on the interface. No restrictions occur for the interface IPv6 link-local address other than that mentioned for the RAs. Other protocols continue to receive and send packets to this address.

HSRP uses a priority mechanism to determine which HSRP configured router is to be the default active router. To configure a router as the active router, you assign it a priority that is higher than the priority of all the other HSRP-configured routers. The default priority is 100, so if you configure just one router to have a higher priority, that router will be the default active router.

For more information see the "Configuring First Hop Redundancy Protocols in IPv6" chapter of the *Cisco IOS IPv6 Configuration Guide*.

HSRP Messages and States

Devices configured with HSRP exchange three types of multicast messages:

- Coup—When a standby device wants to assume the function of the active device, it sends a coup message.
- Hello—The hello message conveys to other HSRP device the HSRP priority and state information of the device.
- Resign—A device that is the active device sends this message when it is about to shut down or when a device that has a higher priority sends a hello or coup message.

At any time, a device configured with HSRP is in one of the following states:

- Active—The device is performing packet-transfer functions.
- Init or Disabled—The device is not yet ready or able to participate in HSRP, possibly because the associated interface is not up. HSRP groups configured on other devices on the network that are learned via snooping are displayed as being in the Init state. Locally configured groups with an interface that is down or groups without a specified interface IP address appear in the Init state.

- Learn—The device has not determined the virtual IP address and has not yet seen an authenticated hello message from the active device. In this state, the device still waits to hear from the active device.
- Listen—The device is receiving hello messages.
- Speak—The device is sending and receiving hello messages.
- Standby—The device is prepared to assume packet-transfer functions if the active device fails.

HSRP uses logging Level 5 for syslog messages related to HSRP state changes to allow logging of an event without filling up the syslog buffer on the device with low-priority Level 6 messaging.

HSRP Group Linking to IP Redundancy Clients

HSRP provides stateless redundancy for IP routing. HSRP by itself is limited to maintaining its own state. Linking an IP redundancy client to an HSRP group provides a mechanism that allows HSRP to provide a service to client applications so they can implement stateful failover.

IP redundancy clients are other Cisco IOS processes or applications that use HSRP to provide or withhold a service or resource dependent upon the state of the group.

HSRP groups have a default name of **hsrp-***interface-group* so specifying a group name is optional. For example, Group 1 on Ethernet0/0 has a default group name of "hsrp-Et0/0-1."

HSRP Object Tracking

Object tracking separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by any other process as well as HSRP. The priority of a device can change dynamically when it has been configured for object tracking and the object that is being tracked goes down. Examples of objects that can be tracked are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, the HSRP priority is reduced.

A client process such as HSRP, Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) can register its interest in tracking objects and then be notified when the tracked object changes state.

For more information about object tracking, see the "Configuring Enhanced Object Tracking" document.

HSRP Group Shutdown

The FHRP—HSRP Group Shutdown feature enables you to configure an HSRP group to become disabled (its state changed to Init) instead of having its priority decremented when a tracked object goes down. Use the **standby track** command with the **shutdown** keyword to configure HSRP group shutdown.

If an object is already being tracked by an HSRP group, you cannot change the configuration to use the HSRP Group Shutdown feature. You must first remove the tracking configuration using the **no standby track** command and then reconfigure it using the **standby track** command with the **shutdown** keyword.

HSRP Support for ICMP Redirect Messages

By default, HSRP filtering of Internet Control Message Protocol (ICMP) redirect messages is enabled on devices running HSRP.

ICMP is a network layer Internet protocol that provides message packets to report errors and other information relevant to IP processing. ICMP can send error packets to a host and can send redirect packets to a host.

When HSRP is running, preventing hosts from discovering the interface (or real) IP addresses of devices in the HSRP group is important. If a host is redirected by ICMP to the real IP address of a device, and that device later fails, then packets from the host will be lost.

ICMP redirect messages are automatically enabled on interfaces configured with HSRP. This functionality works by filtering outgoing ICMP redirect messages through HSRP, where the next hop IP address may be changed to an HSRP virtual IP address.

ICMP Redirects to Active HSRP Devices

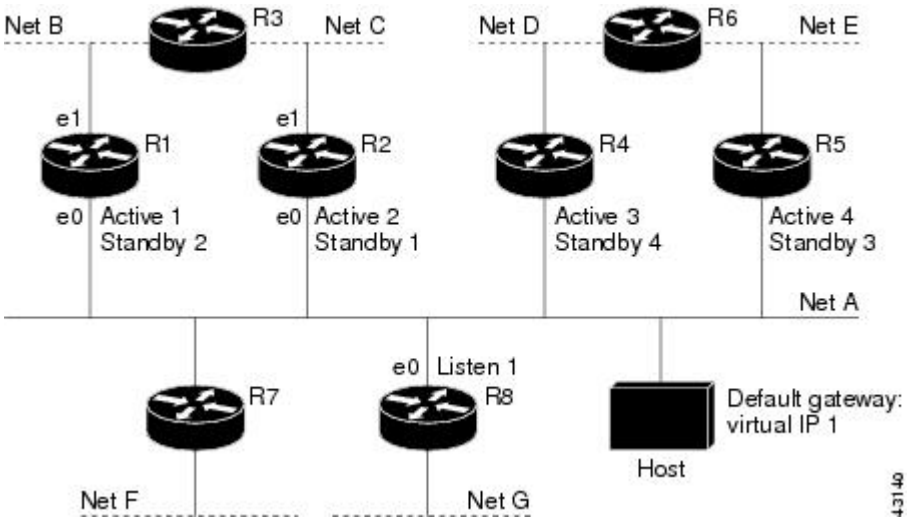
The next-hop IP address is compared to the list of active HSRP devices on that network; if a match is found, then the real next-hop IP address is replaced with a corresponding virtual IP address and the redirect message is allowed to continue.

If no match is found, then the ICMP redirect message is sent only if the device corresponding to the new next hop IP address is not running HSRP. Redirects to passive HSRP devices are not allowed (a passive HSRP device is a device running HSRP, but which contains no active HSRP groups on the interface).

For optimal operation, every device in a network that is running HSRP should contain at least one active HSRP group on an interface to that network. Every HSRP device need not be a member of the same group. Each HSRP device will snoop on all HSRP packets on the network to maintain a list of active devices (virtual IP addresses versus real IP addresses).

Consider the network shown in the figure below, which supports the HSRP ICMP redirection filter.

Figure 11: Network Supporting the HSRP ICMP Redirection Filter



If the host wants to send a packet to another host on Net D, then it first sends it to its default gateway, the virtual IP address of HSRP group 1.

The following is the packet received from the host:

```
dest MAC         = HSRP group 1 virtual MAC
source MAC       = Host MAC
```

```
dest IP          = host-on-netD IP
source IP        = Host IP
```

Device R1 receives this packet and determines that device R4 can provide a better path to Net D, so it prepares to send a redirect message that will redirect the host to the real IP address of device R4 (because only real IP addresses are in its routing table).

The following is the initial ICMP redirect message sent by device R1:

```
dest MAC         = Host MAC
source MAC       = router R1 MAC
dest IP          = Host IP
source IP        = router R1 IP
gateway to use   = router R4 IP
```

Before this redirect occurs, the HSRP process of device R1 determines that device R4 is the active HSRP device for group 3, so it changes the next hop in the redirect message from the real IP address of device R4 to the virtual IP address of group 3. Furthermore, it determines from the destination MAC address of the packet that triggered the redirect message that the host used the virtual IP address of group 1 as its gateway, so it changes the source IP address of the redirect message to the virtual IP address of group 1.

The modified ICMP redirect message showing the two modified fields (*) is as follows:

```
dest MAC         = Host MAC
source MAC       = router R1 MAC
dest IP          = Host IP
source IP*       = HSRP group 1 virtual IP
gateway to use*  = HSRP group 3 virtual IP
```

This second modification is necessary because hosts compare the source IP address of the ICMP redirect message with their default gateway. If these addresses do not match, the ICMP redirect message is ignored. The routing table of the host now consists of the default gateway, virtual IP address of group 1, and a route to Net D through the virtual IP address of group 3.

ICMP Redirects to Passive HSRP Devices

ICMP redirects to passive HSRP devices are not permitted. Redundancy may be lost if hosts learn the real IP addresses of HSRP devices.

In the "Network Supporting the HSRP ICMP Redirection Filter" figure, redirection to device R8 is not allowed because R8 is a passive HSRP device. In this case, packets from the host to Net D will first go to device R1 and then be forwarded to device R4; that is, they will traverse the network twice.

A network configuration with passive HSRP devices is considered a misconfiguration. For HSRP ICMP redirection to operate optimally, every device on the network that is running HSRP should contain at least one active HSRP group.

ICMP Redirects to Non-HSRP Devices

ICMP redirects to devices not running HSRP on their local interface are permitted. No redundancy is lost if hosts learn the real IP address of non-HSRP devices.

In the "Network Supporting the HSRP ICMP Redirection Filter" figure, redirection to device R7 is allowed because R7 is not running HSRP. In this case, the next hop IP address is unchanged. The source IP address is changed dependent upon the destination MAC address of the original packet. You can specify the **no standby redirect unknown** command to stop these redirects from being sent.

Passive HSRP Advertisement Messages

Passive HSRP devices send out HSRP advertisement messages both periodically and when entering or leaving the passive state. Thus, all HSRP devices can determine the HSRP group state of any HSRP device on the network. These advertisements inform other HSRP devices on the network of the HSRP interface state, as follows:

- Active—Interface has at least one active group. A single advertisement is sent out when the first group becomes active.
- Dormant—Interface has no HSRP groups. A single advertisement is sent once when the last group is removed.
- Passive—Interface has at least one nonactive group and no active groups. Advertisements are sent out periodically.

You can adjust the advertisement interval and hold-down time using the **standby redirect timers** command.

ICMP Redirects Not Sent

If the HSRP device cannot uniquely determine the IP address used by the host when it sends the packet that caused the redirect, the redirect message will not be sent. The device uses the destination MAC address in the original packet to make this determination. In certain configurations, such as the use of the **standby use-bia** interface configuration command specified on an interface, redirects cannot be sent. In this case, the HSRP groups use the interface MAC address as their virtual MAC address. The device now cannot determine if the default gateway of the host is the real IP address or one of the HSRP virtual IP addresses that are active on the interface.

The IP source address of an ICMP packet must match the gateway address used by the host in the packet that triggered the ICMP packet, otherwise the host will reject the ICMP redirect packet. An HSRP device uses the destination MAC address to determine the gateway IP address of the host. If the HSRP device is using the same MAC address for multiple IP addresses, uniquely determining the gateway IP address of the host is not possible, and the redirect message is not sent.

The following is sample output from the **debug standby events icmp** EXEC command if HSRP could not uniquely determine the gateway used by the host:

```
10:43:08: HSRP: ICMP redirect not sent to 10.0.0.4 for dest 10.0.1.2
10:43:08: HSRP: could not uniquely determine IP address for mac 00d0.bbd3.bc22
```

HSRP Support for MPLS VPNs

HSRP support for a Multiprotocol Label Switching (MPLS) VPN interface is useful when an Ethernet LAN is connected between two provider edge (PE) devices with either of the following conditions:

- A customer edge (CE) device with a default route to the HSRP virtual IP address
- One or more hosts with the HSRP virtual IP address configured as the default gateway

Each VPN is associated with one or more VPN routing and forwarding (VRF) instances. A VRF consists of the following elements:

- IP routing table

- Cisco Express Forwarding table
- Set of interfaces that use the Cisco Express Forwarding forwarding table
- Set of rules and routing protocol parameters to control the information in the routing tables

VPN routing information is stored in the IP routing table and the Cisco Express Forwarding table for each VRF. A separate set of routing and Cisco Express Forwarding tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN and also prevent packets that are outside a VPN from being forwarded to a device within the VPN.

HSRP adds ARP entries and IP hash table entries (aliases) using the default routing table instance. However, a different routing table instance is used when VRF forwarding is configured on an interface, causing ARP and ICMP echo requests for the HSRP virtual IP address to fail.

HSRP support for MPLS VPNs ensures that the HSRP virtual IP address is added to the correct IP routing table and not to the default routing table.

HSRP Multiple Group Optimization

The configuration of many hundreds of subinterfaces on the same physical interface, with each subinterface having its own HSRP group, can cause the processes of negotiation and maintenance of multiple HSRP groups to have a detrimental impact on network traffic and CPU utilization.

Only one HSRP group is required on a physical interface for the purposes of electing active and standby devices. This group is known as the *primary* group. Other HSRP groups may be created on each subinterface and linked to the primary group via the group name. These linked HSRP groups are known as *client* or *subordinate* groups.

The HSRP group state of the client groups follows that of the primary group. Client groups do not participate in any sort of device election mechanism.

Client groups send periodic messages in order to refresh their virtual MAC addresses in switches and learning bridges. The refresh message may be sent at a much lower frequency compared with the protocol election messages sent by the primary group.

HSRP—ISSU

The In Service Software Upgrade (ISSU) process allows Cisco software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco software to be modified while packet forwarding continues, which increases network availability and reduces downtime caused by planned software upgrades.

For detailed information about ISSU, see the *Cisco IOS In Service Software Upgrade Process* document in the *High Availability Configuration Guide*.

SSO HSRP

SSO HSRP alters the behavior of HSRP when a device with redundant Route Processors (RPs) is configured for stateful switchover (SSO) redundancy mode. When an RP is active and the other RP is standby, SSO enables the standby RP to take over if the active RP fails.

With this functionality, HSRP SSO information is synchronized to the standby RP, allowing traffic that is sent using the HSRP virtual IP address to be continuously forwarded during a switchover without a loss of

data or a path change. Additionally, if both RPs fail on the active HSRP device, then the standby HSRP device takes over as the active HSRP device.

The feature is enabled by default when the redundancy mode of operation is set to SSO.

SSO Dual-Route Processors and Cisco Nonstop Forwarding

SSO functions in networking devices (usually edge devices) that support dual RPs. SSO provides RP redundancy by establishing one of the RPs as the active processor and the other RP as the standby processor. SSO also synchronizes critical state information between the RPs so that network state information is dynamically maintained between RPs.

SSO is generally used with Cisco nonstop forwarding (NSF). Cisco NSF enables forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With NSF, users are less likely to experience service outages.

HSRP and SSO Working Together

The SSO HSRP feature enables the Cisco IOS HSRP subsystem software to detect that a standby RP is installed and the system is configured in SSO redundancy mode. Further, if the active RP fails, no change occurs to the HSRP group itself and traffic continues to be forwarded through the current active gateway device.

Prior to introduction of the SSO HSRP feature, when the primary RP of the active device failed, it would stop participating in the HSRP group and trigger another switch in the group to take over as the active HSRP switch.

SSO HSRP is required to preserve the forwarding path for traffic destined to the HSRP virtual IP address through an RP switchover.

Configuring SSO on the edge device enables the traffic on the Ethernet links to continue during an RP failover without the Ethernet traffic switching over to an HSRP standby device (and then back, if preemption is enabled).

Note You may want to disable SSO HSRP by using the **no standby sso** command if you have LAN segments that should switch HSRP traffic to a redundant device while SSO maintains traffic flow for other connections.

HSRP BFD Peering

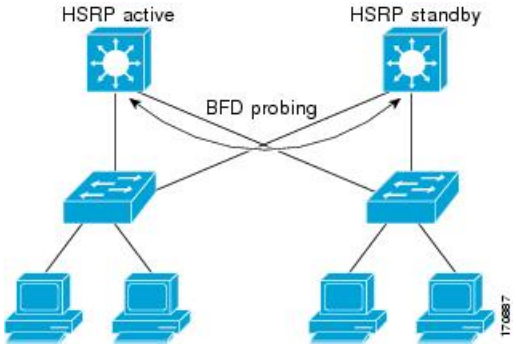
The HSRP BFD Peering feature introduces Bidirectional Forwarding Detection (BFD) in the Hot Standby Router Protocol (HSRP) group member health monitoring system. HSRP supports BFD as a part of the HSRP group member health monitoring system. Without BFD, HSRP runs as a process in a multiprocess system and cannot be guaranteed to be scheduled in time to service large numbers of groups with hello and hold timers, in milliseconds. BFD runs as a pseudopreemptive process and can therefore be guaranteed to run when required. Only one BFD session between two devices can provide early failover notification for multiple HSRP groups.

This feature is enabled by default. The HSRP standby device learns the real IP address of the HSRP active device from the HSRP hello messages. The standby device registers as a BFD client and asks to be notified if the active device becomes unavailable. When BFD determines that the connections between standby and active devices has failed, it will notify HSRP on the standby device which will immediately take over as the active device.

BFD provides a low-overhead, short-duration method of detecting failures in the forwarding path between two adjacent devices, including the interfaces, data links, and forwarding planes. BFD is a detection protocol that you enable at the interface and routing protocol levels. Cisco supports the BFD asynchronous mode, which depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between devices. Therefore, to create a BFD session, you must configure BFD on both systems (or BFD peers). When BFD is enabled on the interfaces and at the device level for HSRP, a BFD session is created, BFD timers are negotiated, and the BFD peers will begin to send BFD control packets to each other at the negotiated interval.

BFD provides fast BFD peer failure detection times independently of all media types, encapsulations, topologies, and routing protocols such as, Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), Hot Standby Router Protocol (HSRP), Intermediate System To Intermediate System (IS-IS), and Open Shortest Path First (OSPF). By sending rapid failure detection notices to the routing protocols in the local device to initiate the routing table recalculation process, BFD contributes to greatly reduce overall network convergence time. The figure below shows a simple network with two devices running HSRP and BFD.

Figure 12: HSRP BFD Peering



For more information about BFD, see the *IP Routing: BFD Configuration Guide*.

HSRP MIB Traps

HSRP MIB supports Simple Network Management Protocol (SNMP) Get operations, to allow network devices to get reports about HSRP groups in a network from the network management station.

Enabling HSRP MIB trap support is performed through the CLI, and the MIB is used for getting the reports. A trap notifies the network management station when a device leaves or enters the active or standby state. When an entry is configured from the CLI, the RowStatus for that group in the MIB immediately goes to the active state.

Cisco software supports a read-only version of the MIB, and set operations are not supported.

This functionality supports four MIB tables, as follows:

- cHsrpGrpEntry table defined in CISCO-HSRP-MIB.my
- cHsrpExtIfTrackedEntry, defined in CISCO-HSRP-EXT-MIB.my
- cHsrpExtSecAddrEntry, defined in CISCO-HSRP-EXT-MIB.my
- cHsrpExtIfEntry defined in CISCO-HSRP-EXT-MIB.my

The cHsrpGrpEntry table consists of all the group information defined in RFC 2281, *Cisco Hot Standby Router Protocol*; the other tables consist of the Cisco extensions to RFC 2281, which are defined in CISCO-HSRP-EXT-MIB.my.

How to Configure HSRP

Enabling HSRP

Perform this task to enable HSRP.

The **standby ip** interface configuration command activates HSRP on the configured interface. If an IP address is specified, that address is used as the virtual IP address for the Hot Standby group. For HSRP to elect a designated device, you must configure the virtual IP address for at least one of the devices in the group; it can be learned on the other devices in the group.

Before you begin

You can configure many attributes in HSRP such as authentication, timers, priority, and preemption. You should configure the attributes before enabling the HSRP group. This practice avoids authentication error messages and unexpected state changes in other routers that can occur if the group is enabled first and then there is a long enough delay (one or two hold times) before the other attribues are configured.

We recommend that you always specify an HSRP IP address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
6. **end**
7. **show standby** [**all**] [**brief**]
8. **show standby** *type number* [*group-number* | **all**] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 172.16.6.5 255.255.255.0`	Configures an IP address for an interface.
Step 5	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 172.16.6.100`	Activates HSRP. • If you do not configure a group number, the default group number is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2. • The value for the *ip-address* argument is the virtual IP address of the virtual device. For HSRP to elect a designated device, you must configure the virtual IP address for at least one of the devices in the group; it can be learned on the other devices in the group.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show standby** [**all**] [**brief**] **Example:** `Device# show standby`	(Optional) Displays HSRP information. • This command displays information for each group. The **all** option displays groups that are learned or that do not have the **standby ip** command configured.
Step 8	**show standby** *type number* [*group-number* | **all**] [**brief**] **Example:** `Device# show standby GigabitEthernet 0`	(Optional) Displays HSRP information about specific groups or interfaces.

Delaying the Initialization of HSRP on an Interface

The **standby delay** command is used to delay HSRP initialization either after a reload and/or after an interface comes up. This configuration allows the interface and device time to settle down after the interface up event and helps prevent HSRP state flapping.

We recommend that you use the **standby minimum reload** command if the **standby timers** command is configured in milliseconds or if HSRP is configured on a VLAN interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **standby delay minimum** *min-seconds* **reload** *reload-seconds*
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
7. **end**
8. **show standby delay** [*typenumber*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies an IP address for an interface.
Step 5	**standby delay minimum** *min-seconds* **reload** *reload-seconds* **Example:** `Device(config-if)# standby delay minimum 30 reload 60`	(Optional) Configures the delay period before the initialization of HSRP groups. • The *min-seconds* value is the minimum time (in seconds) to delay HSRP group initialization after an interface comes up. This minimum delay period applies to all subsequent interface events. • The *reload-seconds* value is the time period to delay after the device has reloaded. This delay period applies only to the first interface-up event after the device has reloaded. **Note** The recommended *min-seconds* value is 30 and the recommended *reload-seconds* value is 60.

	Command or Action	Purpose
Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3 255.255.255.0`	Activates HSRP.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 8	**show standby delay** [*typenumber*] **Example:** `Device# show standby delay`	(Optional) Displays HSRP information about delay periods.

Configuring HSRP Priority and Preemption

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **standby** [*group-number*] **priority** *priority*
6. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*]
7. **standby** [*group-number*] **ip** *ip-address* [**secondary**]]
8. **end**
9. **show standby** [**all**] [**brief**]
10. **show standby** *type number* [*group-number* | **all**] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies an IP address for an interface.
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority. • The default priority is 100.
Step 6	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*] **Example:** `Device(config-if)# standby 1 preempt delay minimum 380`	Configures HSRP preemption and preemption delay. • The default delay period is 0 seconds; if the device wants to preempt, it will do so immediately. By default, the device that comes up later becomes the standby.
Step 7	**standby** [*group-number*] **ip** *ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3 255.255.255.0`	Activates HSRP.
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 9	**show standby** [**all**] [**brief**] **Example:** `Device# show standby`	(Optional) Displays HSRP information. • This command displays information for each group. The **all** option displays groups that are learned or that do not have the **standby ip** command configured.
Step 10	**show standby** *type number* [*group-number* | **all**] [**brief**] **Example:** `Device# show standby GigabitEthernet 0/0/0`	(Optional) Displays HSRP information about specific groups or interfaces.

Configuring HSRP Object Tracking

Perform this task to configure HSRP to track an object and change the HSRP priority based on the state of the object.

Each tracked object is identified by a unique number that is specified on the tracking CLI. Client processes use this number to track a specific object.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **interface** *type number* {**line-protocol** | **ip routing**}
4. **exit**
5. **interface** *type number*
6. **standby** [*group-number*] **track** *object-number* [**decrement** *priority-decrement*] [**shutdown**]
7. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
8. **end**
9. **show track** [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**track** *object-number* **interface** *type number* {**line-protocol**	**ip routing**} **Example:** `Device(config)# track 100 interface GigabitEthernet 0/0/0 line-protocol`	Configures an interface to be tracked and enters tracking configuration mode.
Step 4	**exit** **Example:** `Device(config-track)# exit`	Returns to global configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 6	**standby** [*group-number*] **track** *object-number* [**decrement** *priority-decrement*] [**shutdown**] **Example:** `Device(config-if)# standby 1 track 100 decrement 20`	Configures HSRP to track an object and change the Hot Standby priority on the basis of the state of the object. • By default, the priority of the device is decreased by 10 if a tracked object goes down. Use the **decrement** *priority-decrement* keyword and argument combination to change the default behavior. • When multiple tracked objects are down and *priority-decrement* values have been configured, these configured priority decrements are cumulative. If tracked objects are down, but none of them were configured with priority decrements, the default decrement is 10 and it is cumulative. • Use the **shutdown** keyword to disable the HRSP group on the device when the tracked object goes down. **Note** If an object is already being tracked by an HSRP group, you cannot change the configuration to use the HSRP Group Shutdown feature. You must first remove the tracking configuration using the **no standby track** command and then reconfigure it using the **standby track** command with the **shutdown** keyword.
Step 7	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.10.10.0`	Activates HSRP. • The default group number is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2.
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 9	**show track** [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**] **Example:** `Device# show track 100 interface`	Displays tracking information.

Configuring HSRP MD5 Authentication Using a Key String

Note Text authentication cannot be combined with MD5 authentication for an HSRP group at any one time. When MD5 authentication is configured, the text authentication field in HSRP hello messages is set to all zeroes on transmit and ignored on receipt, provided the receiving device also has MD5 authentication enabled.

Note If you are changing a key string in a group of devices, change the active device last to prevent any HSRP state change. The active device should have its key string changed no later than one hold-time period, specified by the **standy timers** interface configuration command, after the nonactive devices. This procedure ensures that the nonactive devices do not time out the active device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **terminal interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby** [*group-number*] **priority** *priority*
6. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*]
7. **standby** [*group-number*] **authentication md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*]
8. **standby** [*group-number*] **ip** [*ip-address*] [**secondary**]]
9. Repeat Steps 1 through 8 on each device that will communicate.
10. **end**
11. **show standby**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**terminal interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 6	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption.
Step 7	**standby** [*group-number*] **authentication md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] **Example:** `Device(config-if)# standby 1 authentication md5 key-string d00b4r987654321a timeout 30`	Configures an authentication string for HSRP MD5 authentication. • The *key* argument can be up to 64 characters in length. We recommended that at least 16 characters be used. • No prefix to the *key* argument or specifying **0** means the key will be unencrypted. • Specifying **7** means the key will be encrypted. The key-string authentication key will automatically be encrypted if the **service password-encryption** global configuration command is enabled. • The **timeout** value is the period of time that the old key string will be accepted to allow configuration of all routers in a group with a new key.
Step 8	**standby** [*group-number*] **ip** [*ip-address*] [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3`	Activates HSRP.
Step 9	Repeat Steps 1 through 8 on each device that will communicate.	—
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 11	**show standby** **Example:**	(Optional) Displays HSRP information.

	Command or Action	Purpose
	`Device# show standby`	• Use this command to verify your configuration. The key string or key chain will be displayed if configured.

Configuring HSRP MD5 Authentication Using a Key Chain

Perform this task to configure HSRP MD5 authentication using a key chain. Key chains allow a different key string to be used at different times according to the key chain configuration. HSRP will query the appropriate key chain to obtain the current live key and key ID for the specified key chain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*
5. **key-string** *string*
6. **exit**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask* [**secondary**]
10. **standby** [*group-number*] **priority** *priority*
11. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*]
12. **standby** [*group-number*] **authentication md5 key-chain** *key-chain-name*
13. **standby** [*group-number*] **ip** [*ip-address* [*secondary*]]
14. Repeat Steps 1 through 12 on each device that will communicate.
15. **end**
16. **show standby**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain hsrp1`	Enables authentication for routing protocols, identifies a group of authentication keys, and enters key-chain configuration mode.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 100`	Identifies an authentication key on a key chain and enters key-chain key configuration mode. • The value for the*key-id* argument must be a number.
Step 5	**key-string** *string* **Example:** `Device(config-keychain-key)# key-string mno172`	Specifies the authentication string for a key. • The value for the *string* argument can be 1 to 80 uppercase or lowercase alphanumeric characters; the first character cannot be a numeral
Step 6	**exit** **Example:** `Device(config-keychain-key)# exit`	Returns to key-chain configuration mode.
Step 7	**exit** **Example:** `Device(config-keychain)# exit`	Returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.21.8.32 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 10	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 11	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption.

	Command or Action	Purpose
Step 12	**standby** [*group-number*] **authentication md5 key-chain** *key-chain-name* **Example:** `Device(config-if)# standby 1 authentication md5 key-chain hsrp1`	Configures an authentication MD5 key chain for HSRP MD5 authentication. • The key chain name must match the name specified in Step 3.
Step 13	**standby** [*group-number*] **ip** [*ip-address* [*secondary*]] **Example:** `Device(config-if)# standby 1 ip 10.21.8.12`	Activates HSRP.
Step 14	Repeat Steps 1 through 12 on each device that will communicate.	—
Step 15	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 16	**show standby** **Example:** `Device# show standby`	(Optional) Displays HSRP information. • Use this command to verify your configuration. The key string or key chain will be displayed if configured.

Troubleshooting HSRP MD5 Authentication

Perform this task if HSRP MD5 authentication is not operating correctly.

SUMMARY STEPS

1. **enable**
2. **debug standby errors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug standby errors** **Example:** `Device# debug standby errors`	Displays error messages related to HSRP. • Error messages will be displayed for each packet that fails to authenticate, so use this command with care.

Examples

In the following example, Device A has MD5 text string authentication configured, but Device B has the default text authentication:

```
Device# debug standby errors

A:Jun 16 12:14:50.337:HSRP:Et0/1 Grp 0 Auth failed for Hello pkt from 10.21.0.5, MD5 confgd
 but no tlv
B:Jun 16 12:16:34.287:HSRP:Et0/1 Grp 0 Auth failed for Hello pkt from 10.21.0.4, Text auth
 failed
```

In the following example, both Device A and Device B have different MD5 authentication strings:

```
Device# debug standby errors

A:Jun 16 12:19:26.335:HSRP:Et0/1 Grp 0 Auth failed for Hello pkt from 10.21.0.5, MD5 auth
failed
B:Jun 16 12:18:46.280:HSRP:Et0/1 Grp 0 Auth failed for Hello pkt from 10.21.0.4, MD5 auth
failed
```

Configuring HSRP Text Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby** [*group-number*] **priority** *priority*
6. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*]
7. **standby** [*group-number*] **authentication text** *string*
8. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
9. Repeat Steps 1 through 8 on each device that will communicate.
10. **end**
11. **show standby**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 6	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption.
Step 7	**standby** [*group-number*] **authentication text** *string* **Example:** `Device(config-if)# standby 1 authentication text authentication1`	Configures an authentication string for HSRP text authentication. • The default string is cisco.
Step 8	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3`	Activates HSRP.
Step 9	Repeat Steps 1 through 8 on each device that will communicate.	--
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 11	**show standby** **Example:** `Device# show standby`	(Optional) Displays HSRP information. • Use this command to verify your configuration. The key string or key chain will be displayed if configured.

Configuring HSRP Timers

Note We recommend configuring a minimum hello-time value of 250 milliseconds and a minimum hold-time value of 800 milliseconds.

You can use the **standby delay** command to allow the interface to come up completely before HSRP initializes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]]
5. **standby** [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime*
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Gigabit Ethernet 0/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**]] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby** [*group-number*] **timers** [**msec**] *hellotime* [**msec**] *holdtime* **Example:** `Device(config-if)# standby 1 timers 5 15`	Configures the time between hello packets and the time before other devices declare the active Hot Standby or standby device to be down.

	Command or Action	Purpose
Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3`	Activates HSRP.

Configuring an HSRP MAC Refresh Interval

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby mac-refresh** *seconds*
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby mac-refresh** *seconds* **Example:** `Device(config-if)# standby mac-refresh 100`	Changes the interval at which packets are sent to refresh the MAC cache when HSRP is running over FDDI. • This command applies to HSRP running over FDDI only.

	Command or Action	Purpose
Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3`	Activates HSRP.

Configuring Multiple HSRP Groups for Load Balancing

Perform this task to configure multiple HSRP groups for load balancing.

Multiple HSRP groups enable redundancy and load-sharing within networks and allow redundant devices to be more fully utilized. A device actively forwarding traffic for one HSRP group can be in standby or in the listen state for another group.

If two devices are used, then Device A would be configured as active for group 1 and standby for group 2. Device B would be standby for group 1 and active for group 2. Fifty percent of the hosts on the LAN would be configured with the virtual IP address of group 1 and the remaining hosts would be configured with the virtual IP address of group 2. See the Example: Configuring Multiple HSRP Groups for Load Balancing for a diagram and configuration example.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby** [*group-number*] **priority** *priority*
6. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *delay*]
7. **standby** [*group-number*] **ip** [*ip-address*] **secondary**]
8. On the same device, repeat Steps 5 through 7 to configure the device attributes for different standby groups.
9. **exit**
10. Repeat Steps 3 through 9 on another device.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 6	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *delay*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption.
Step 7	**standby** [*group-number*] **ip** [*ip-address*] **secondary**] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3`	Activates HSRP.
Step 8	On the same device, repeat Steps 5 through 7 to configure the device attributes for different standby groups.	For example, Device A can be configured as an active device for group 1 and be configured as an active or standby device for another HSRP group with different priority and preemption values.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 10	Repeat Steps 3 through 9 on another device.	Configures multiple HSRP and enables load balancing on another device.

Improving CPU and Network Performance with HSRP Multiple Group Optimization

Perform this task to configure multiple HSRP client groups.

The **standby follow** command configures an HSRP group to become a subordinate of another HSRP group.

HSRP client groups follow the master HSRP with a slight, random delay so that all client groups do not change at the same time.

Use the **standby mac-refresh** *seconds* command to directly change the HSRP client group refresh interval. The default interval is 10 seconds and can be configured to as much as 255 seconds.

Note

- Client or subordinate groups must be on the same physical interface as the master group.
- A client group takes its state from the group it is following. Therefore, the client group does not use its timer, priority, or preemption settings. A warning is displayed if these settings are configured on a client group:

```
Device(config-if)# standby 1 priority 110
%Warning: This setting has no effect while following another group.
Device(config-if)# standby 1 timers 5 15
    % Warning: This setting has no effect while following another group.
Device(config-if)# standby 1 preempt delay minimum 300
    % Warning: This setting has no effect while following another group.
```

Before you begin

Configure the HSRP master group using the steps in the Configuring Multiple HSRP Groups for Load Balancing section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby mac-refresh** *seconds*
6. **standby** *group-number* **follow** *group-name*
7. **exit**
8. Repeat Steps 3 through 6 to configure additional HSRP client groups.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby mac-refresh** *seconds* **Example:** `Device(config-if)# standby mac-refresh 30`	Configures the HSRP client group refresh interval.
Step 6	**standby** *group-number* **follow** *group-name* **Example:** `Device(config-if)# standby 1 follow HSRP1`	Configures an HSRP group as a client group.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	Repeat Steps 3 through 6 to configure additional HSRP client groups.	Configures multiple HSRP client groups.

Enabling HSRP Support for ICMP Redirect Messages

By default, HSRP filtering of ICMP redirect messages is enabled on devices running HSRP. Perform this task to reenable this feature on your device if it is disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **standby redirect** [**timers** *advertisement holddown*] [**unknown**]
5. **end**
6. **show standby redirect** [*ip-address*] [*interface-type interface-number*] [**active**] [**passive**] [**timers**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**standby redirect** [**timers** *advertisement holddown*] [**unknown**] **Example:** `Device(config-if)# standby redirect`	Enables HSRP filtering of ICMP redirect messages. • You can also use this command in global configuration mode, which enables HSRP filtering of ICMP redirect messages on all interfaces configured for HSRP.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show standby redirect** [*ip-address*] [*interface-type interface-number*] [**active**] [**passive**] [**timers**] **Example:** `Device# show standby redirect`	(Optional) Displays ICMP redirect information on interfaces configured with HSRP.

Configuring HSRP Virtual MAC Addresses or BIA MAC Addresses

Note You cannot use the **standby use-bia** and **standby mac-address** commands in the same configuration; they are mutually exclusive.

The **standby use-bia** command has the following disadvantages:

- When a device becomes active the virtual IP address is moved to a different MAC address. The newly active device sends a gratuitous ARP response, but not all host implementations handle the gratuitous ARP correctly.
- Proxy ARP does not function when the **standby use-bia** command is configured. A standby device cannot cover for the lost proxy ARP database of the failed device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. Enter one of the following commands:
   - **standby** [*group-number*] **mac-address** *mac-address*
   - or
   - **standby use-bia** [**scope interface**]
   - or
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 172.16.6.5 255.255.255.0`	Configures an IP address for an interface.

Step 5	Enter one of the following commands:
	• **standby** [*group-number*] **mac-address** *mac-address*
	• or
	• **standby use-bia** [**scope interface**]
	• or
	Example:
	`Device(config-if)# standby 1 mac-address 5000.1000.1060`
	Example:
	`Device(config-if)# standby use-bia`
		Specifies a virtual MAC address for HSRP.
		• This command cannot be used on a Token Ring interface.
		or
		Configures HSRP to use the burned-in address of the interface as its virtual MAC address.
		• The **scope interface** keywords specify that the command is configured just for the subinterface on which it was entered, instead of the major interface.

Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 172.16.6.100`	Activates HSRP.

Linking IP Redundancy Clients to HSRP Groups

Before you begin

Within the client application, you must first specify the same name as configured in the **standby name** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **standby** [*group-number*] **name** [*redundancy-name*]
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies an IP address for an interface.
Step 5	**standby** [*group-number*] **name** [*redundancy-name*] **Example:** `Device(config-if)# standby 1 name HSRP-1`	Configures the name of the standby group. • HSRP groups have a default name of **hsrp**-*interface-group* so specifying a group name is optional.
Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.11`	Activates HSRP.

Changing to HSRP Version 2

HSRP version 2 was introduced to prepare for further enhancements and to expand the capabilities beyond what is possible with HSRP version 1. HSRP version 2 has a different packet format than HSRP version 1.

Note
- HSRP version 2 is not available for ATM interfaces running LAN emulation.
- HSRP version 2 will not interoperate with HSRP version 1. An interface cannot operate both version 1 and version 2 because both versions are mutually exclusive. However, the different versions can be run on different physical interfaces of the same device. You cannot change from version 2 to version 1 if you have configured groups above the group number range allowed for version 1 (0 to 255).

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **standby version** {**1** | **2**}
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
7. **end**
8. **show standby**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface vlan 400`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.10.28.1 255.255.255.0`	Sets an IP address for an interface.
Step 5	**standby version** {**1**	**2**} **Example:** `Device(config-if)# standby version 2`	Changes the HSRP version. **Note** The **standby version 2** command appears in the running configuration as it's a non-default setting. However, the **standby version 1** command doesn't appear because it represents the default HSRP version.
Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 400 ip 10.10.28.5`	Activates HSRP. • The group number range for HSRP version 2 is 0 through 4095. The group number range for HSRP version 1 is 0 through 255.
Step 7	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show standby** **Example:** `Device# show standby`	(Optional) Displays HSRP information. • HSRP version 2 information will be displayed if configured.

Enabling SSO Aware HSRP

The SSO aware HSRP is enabled by default when the redundancy mode is set to SSO. Perform this task to reenable HSRP to be SSO aware if it has been disabled.

Note You may want to disable SSO HSRP by using the **no standby sso** command if you have LAN segments that should switch HSRP traffic to a redundant device while SSO maintains traffic flow for other connections.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **mode sso**
5. **exit**
6. **no standby sso**
7. **standby sso**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**mode sso**	Enables the redundancy mode of operation to SSO.

	Command or Action	Purpose
	Example: `Device(config-red)# mode sso`	• HSRP is SSO aware on interfaces that are configured for HSRP and the standby RP is automatically reset.
Step 5	**exit** **Example:** `Device(config-red)# exit`	Exits redundancy configuration mode.
Step 6	**no standby sso** **Example:** `Device(config)# no standby sso`	Disables HSRP SSO mode for all HSRP groups.
Step 7	**standby sso** **Example:** `Device(config)# standby sso`	Enables the SSO HSRP feature if you have disabled the functionality.
Step 8	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Verifying SSO Aware HSRP

To verify or debug HSRP SSO operation, perform the following steps from the active RP console.

SUMMARY STEPS

1. **show standby**
2. **debug standby events ha**

DETAILED STEPS

Step 1 **show standby**

Use the **show standby** command to display the state of the standby RP, for example:

Example:

```
Device# show standby

GigabitEthernet0/0/0 - Group 1
 State is Active (standby RP)
 Virtual IP address is 10.1.0.7
 Active virtual MAC address is unknown
  Local virtual MAC address is 000a.f3fd.5001 (bia)
 Hello time 1 sec, hold time 3 sec
```

```
 Authentication text "authword"
 Preemption enabled
 Active router is unknown
 Standby router is unknown
 Priority 110 (configured 120)
  Track object 1 state Down decrement 10
 Group name is "name1" (cfgd)
```

Step 2 **debug standby events ha**

Use the**debug standby events ha** command to display the active and standby RPs, for example:

Example:

```
Device# debug standby events ha

!Active RP
*Apr 27 04:13:47.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Listen into sync buffer
*Apr 27 04:13:47.855: HSRP: CF Sync send ok
*Apr 27 04:13:57.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Speak into sync buffer
*Apr 27 04:13:57.855: HSRP: CF Sync send ok
*Apr 27 04:14:07.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Standby into sync buffer
*Apr 27 04:14:07.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Active into sync buffer
*Apr 27 04:14:07.863: HSRP: CF Sync send ok
*Apr 27 04:14:07.867: HSRP: CF Sync send ok
!Standby RP
*Apr 27 04:11:21.011: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:21.011: HSRP: Gi0/0/1 Grp 101 RF sync state Init -> Listen
*Apr 27 04:11:31.011: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:31.011: HSRP: Gi0/0/1 Grp 101 RF sync state Listen -> Speak
*Apr 27 04:11:41.071: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:41.071: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:41.071: HSRP: Gi0/0/1 Grp 101 RF sync state Speak -> Standby
*Apr 27 04:11:41.071: HSRP: Gi0/0/1 Grp 101 RF sync state Standby -> Active
```

Enabling HSRP MIB Traps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps hsrp**
4. **snmp-server host** *host community-string* **hsrp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps hsrp** **Example:** `Device(config)# snmp-server enable traps hsrp`	Enables the device to send SNMP traps and informs, and HSRP notifications.
Step 4	**snmp-server host** *host community-string* **hsrp** **Example:** `Device(config)# snmp-server host myhost.comp.com public hsrp`	Specifies the recipient of an SNMP notification operation, and that HSRP notifications be sent to the host.

Configuring BFD Session Parameters on an Interface

Perform this task to configure Bidirectional Forwarding Detection (BFD) on an interface by setting the baseline BFD session parameters on the interface. Repeat the steps in this task for each interface on which you want to run BFD sessions to BFD neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface FastEthernet 6/0`	
Step 4	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Device(config-if)# bfd interval 50 min_rx 50 multiplier 5`	Enables BFD on the interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode.

Configuring HSRP BFD Peering

Perform this task to enable Hot Standby Router Protocol (HSRP) Bidirectional Forwarding Detection (BFD) peering. Repeat the steps in this task for each interface over which you want to run BFD sessions to HSRP peers.

HSRP supports BFD peering by default. If HSRP BFD peering is disabled, you can reenable it at the device level to enable BFD support globally for all interfaces or you can reenable it on a per-interface basis at the interface level.

Before you begin

Before you proceed with this task:

- HSRP must be running on all participating devices.
- Cisco Express Forwarding must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
7. **standby bfd**
8. **exit**
9. **standby bfd all-interfaces**
10. **exit**
11. **show standby** [**neighbors**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip cef** [**distributed**] **Example:** `Device(config)# ip cef`	Enables Cisco Express Forwarding or distributed Cisco Express Forwarding.
Step 4	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 6/0`	Enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.0.0.11 255.255.255.0`	Configures an IP address for the interface.
Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.11`	Activates HSRP.
Step 7	**standby bfd** **Example:** `Device(config-if)# standby bfd`	(Optional) Enables HSRP support for BFD on the interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 9	**standby bfd all-interfaces** **Example:** `Device(config)# standby bfd all-interfaces`	(Optional) Enables HSRP support for BFD on all interfaces.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 11	**show standby** [**neighbors**] **Example:** `Device# show standby neighbors`	(Optional) Displays information about HSRP support for BFD.

Verifying HSRP BFD Peering

To verify Hot Standby Router Protocol (HSRP) Bidirectional Forwarding Detection (BFD) peering, use any of the following optional commands.

SUMMARY STEPS

1. **show standby**
2. **show standby brief**
3. **show standby neighbors** [*type number*]
4. **show bfd neighbors**
5. **show bfd neighbors details**

DETAILED STEPS

Step 1 show standby

Use the **show standby** command to display HSRP information.

Example:

```
Device# show standby

FastEthernet2/0 - Group 1
  State is Active
    2 state changes, last state change 00:08:06
  Virtual IP address is 10.0.0.11
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.772 secs
  Preemption enabled
  Active router is local
  Standby router is 10.0.0.2, priority 90 (expires in 8.268 sec)
    BFD enabled !
  Priority 110 (configured 110)
    Group name is "hsrp-Fa2/0-1" (default)
```

Step 2 show standby brief

Use the **show standby brief** command to display HSRP standby device information in brief.

Example:

```
Device# show standby brief

Interface   Grp  Pri P State   Active  Standby                   Virtual IP

Et0/0       4    120 P Active  local   172.24.1.2                172.24.1.254
Et1/0       6    120 P Active  local   FE80::A8BB:CCFF:FE00:3401   FE80::5:73FF:FEA0:6
```

Step 3 **show standby neighbors** [*type number*]

Use the **show standby neighbors** command to display information about HSRP peer devices on an interface.

Example:

```
Device1# show standby neighbors

HSRP neighbors on FastEthernet2/0
    10.1.0.22
    No active groups
    Standby groups: 1
    BFD enabled !

Device2# show standby neighbors

HSRP neighbors on FastEthernet2/0
    10.0.0.2
    Active groups: 1
    No standby groups
    BFD enabled !
```

Step 4 **show bfd neighbors**

Use the **show bfd neighbors** command to display a line-by-line listing of existing Bidirectional Forwarding Detection (BFD) adjacencies.

Example:

```
Device# show bfd neighbors

IPv6 Sessions

NeighAddr                              LD/RD         RH/RS     State     Int

FE80::A8BB:CCFF:FE00:3401               4/3          Up        Up        Et1/0
FE80::A8BB:CCFF:FE00:3401               4/3          Up        Up        Et1/0
```

Step 5 **show bfd neighbors details**

Use the **details** keyword to display BFD protocol parameters and timers for each neighbor.

Example:

```
Device# show bfd neighbors details

OurAddr       NeighAddr     LD/RD  RH/RS   Holdown(mult)  State     Int
10.0.0.2      10.0.0.1       5/0    Down      0    (0 )   Down      Fa2/0
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 0, Received Multiplier: 0
Holdown (hits): 0(0), Hello (hits): 1000(55)
Rx Count: 0, Rx Interval (ms) min/max/avg: 0/0/0 last: 3314120 ms ago
```

```
Tx Count: 55, Tx Interval (ms) min/max/avg: 760/1000/872 last: 412 ms ago
Registered protocols: HSRP !
Last packet: Version: 1            - Diagnostic: 0
             State bit: AdminDown  - Demand bit: 0
             Poll bit: 0           - Final bit: 0
             Multiplier: 0         - Length: 0
             My Discr.: 0          - Your Discr.: 0
             Min tx interval: 0    - Min rx interval: 0
             Min Echo interval: 0
```

Configuration Examples for HSRP

Example: Configuring HSRP Priority and Preemption

In the following example, Device A is configured to be the active device for group 1 because it has the higher priority and standby device for group 2. Device B is configured to be the active device for group 2 and standby device for group 1.

Device A Configuration

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 2 priority 95
Device(config-if)# standby 2 preempt
Device(config-if)# standby 2 ip 10.1.0.2
```

Device B Configuration

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.1.0.22 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 2 priority 110
Device(config-if)# standby 2 preempt
Device(config-if)# standby 2 ip 10.1.0.2
```

Example: Configuring HSRP Object Tracking

In the following example, the tracking process is configured to track the IP-routing capability of serial interface 1/0. HSRP on Gigabit Ethernet interface 0/0/0 then registers with the tracking process to be informed of any changes to the IP-routing state of serial interface 1/0. If the IP state on serial interface 1/0 goes down, the priority of the HSRP group is reduced by 10.

If both serial interfaces are operational, Device A will be the HSRP active device because it has the higher priority. However, if IP routing on serial interface 1/0 in Device A fails, the HSRP group priority will be

reduced and Device B will take over as the active device, thus maintaining a default virtual gateway service to hosts on the 10.1.0.0 subnet.

Device A Configuration

```
Device(config)# track 100 interface serial 1/0/0 ip routing
!
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
Device(config-if)# standby 1 ip 10.1.0.1
```

Device B Configuration

```
Device(config)# track 100 interface serial 1/0/0 ip routing
!
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.1.0.22 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
Device(config-if)# standby 1 ip 10.1.0.1
```

Example: Configuring HSRP Group Shutdown

In the following example, the tracking process is configured to track the IP-routing capability of Gigabit Ethernet interface 0/0/0. HSRP on Gigabit Ethernet interface 0/0/1 then registers with the tracking process to be informed of any changes to the IP-routing state of Gigabit Ethernet interface 0/0/0. If the IP state on Gigabit Ethernet interface 0/0/0 goes down, the HSRP group is disabled.

If both Gigabit Ethernet interfaces are operational, Device A will be the HSRP active device because it has the higher priority. However, if IP routing on Gigabit Ethernet interface 0/0/0 in Device A fails, the HSRP group will be disabled and Device B will take over as the active device, thus maintaining a default virtual gateway service to hosts on the 10.1.0.0 subnet.

Device A Configuration

```
Device(config)# track 100 interface GigabitEthernet 0/0/0 ip routing
!
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 shutdown
```

Device B Configuration

```
Device(config)# track 100 interface GigabitEthernet 0/0/0 ip routing
!
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip address 10.1.0.22 255.255.0.0
```

```
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 shutdown
```

If an object is already being tracked by an HSRP group, you cannot change the configuration to use the HSRP Group Shutdown feature. You must first remove the tracking configuration using the **no standby track** command and then reconfigure it using the **standby track** command with the **shutdown** keyword.

The following example shows how to change the configuration of a tracked object to include the HSRP Group Shutdown feature:

```
Device(config)# no standby 1 track 100 decrement 10
Device(config)# standby 1 track 100 shutdown
```

Example: Configuring HSRP MD5 Authentication Using Key Strings

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 authentication md5 key-string 54321098452103ab timeout 30
Device(config-if)# standby 1 ip 10.21.0.10
```

Example: Configuring HSRP MD5 Authentication Using Key Chains

In the following example, HSRP queries the key chain “hsrp1” to obtain the current live key and key ID for the specified key chain:

```
Device(config)# key chain hsrp1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 54321098452103ab
Device(config-keychain-key)# exit
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 authentication md5 key-chain hsrp1
Device(config-if)# standby 1 ip 10.21.0.10
```

Example: Configuring HSRP MD5 Authentication Using Key Strings and Key Chains

The key ID for key-string authentication is always zero. If a key chain is configured with a key ID of zero, then the following configuration will work:

Device 1

```
Device(config)# key chain hsrp1
Device(config-keychain)# key 0
Device(config-keychain-key)# key-string 54321098452103ab
Device(config-keychain-key)# exit
Device(config)# interface GigabitEthernet 0/0/0
```

```
Device(config-if)# standby 1 authentication md5 key-chain hsrp1
Device(config-if)# standby 1 ip 10.21.0.10
```

Device 2

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 authentication md5 key-string 54321098452103ab
Device(config-if)# standby 1 ip 10.21.0.10
```

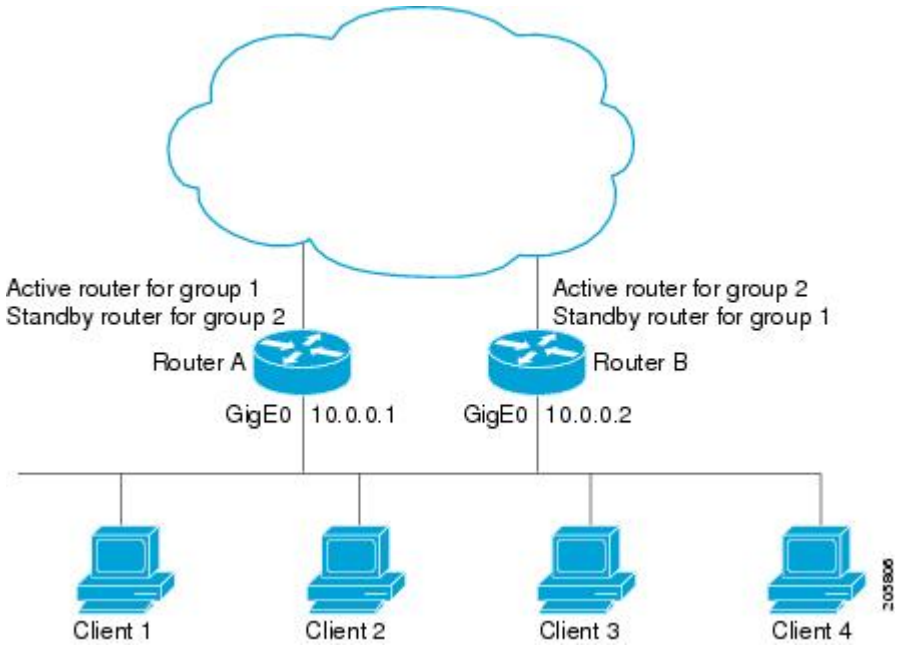
Example: Configuring HSRP Text Authentication

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 authentication text company2
Device(config-if)# standby 1 ip 10.21.0.10
```

Example: Configuring Multiple HSRP Groups for Load Balancing

You can use HSRP or multiple HSRP groups when you configure load sharing. In the figure below, half of the clients are configured for Router A, and half of the clients are configured for Router B. Together, the configuration for Routers A and B establish two Hot Standby groups. For group 1, Router A is the default active router because it has the assigned highest priority, and Router B is the standby router. For group 2, Router B is the default active router because it has the assigned highest priority, and Router A is the standby router. During normal operation, the two routers share the IP traffic load. When either router becomes unavailable, the other router becomes active and assumes the packet-transfer functions of the router that is unavailable. The **standby preempt** interface configuration command is necessary so that if a router goes down and then comes back up, preemption occurs and restores load sharing.

Figure 13: HSRP Load Sharing Example



The following example shows Router A configured as the active router for group 1 with a priority of 110 and Router B configured as the active router for group 2 with a priority of 110. The default priority level is 100. Group 1 uses a virtual IP address of 10.0.0.3 and Group 2 uses a virtual IP address of 10.0.0.4.

Router A Configuration

```
Router(config)# hostname RouterA
!
Router(config)# interface GigabitEthernet 0/0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.0
Router(config-if)# standby 1 priority 110
Router(config-if)# standby 1 preempt
Router(config-if)# standby 1 ip 10.0.0.3
Router(config-if)# standby 2 preempt
Router(config-if)# standby 2 ip 10.0.0.4
```

Router B Configuration

```
Router(config)# hostname RouterB
!
Router(config)# interface GigabitEthernet 0/0/0
Router(config-if)# ip address 10.0.0.2 255.255.255.0
Router(config-if)# standby 1 preempt
Router(config-if)# standby 1 ip 10.0.0.3
Router(config-if)# standby 2 priority 110
Router(config-if)# standby 2 preempt
Router(config-if)# standby 2 ip 10.0.0.4
```

Example: Improving CPU and Network Performance with HSRP Multiple Group Optimization

The following example shows how to configure an HSRP client and primary group:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# no shutdown
Device(config-if)# standby mac-refresh 30
! Client Hello message interval
!
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# no shutdown
Device(config-if)# ip vrf forwarding VRF2
Device(config-if)# ip address 10.0.0.100 255.255.0.0
Device(config-if)# standby 1 ip 10.0.0.254
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 name HSRP1
!Server group
!
Device(config)# interface GigabitEthernet 0/0/2
Device(config-if)# no shutdown
Device(config-if)# ip vrf forwarding VRF3
Device(config-if)# ip address 10.0.0.100 255.255.0.0
Device(config-if)# standby 2 ip 10.0.0.254
Device(config-if)# standby 2 follow HSRP1
! Client group
!
```

```
Device(config)# interface GigabitEthernet 0/0/3
Device(config-if)# no shutdown
Device(config-if)# ip vrf forwarding VRF4
Device(config-if)# ip address 10.0.0.100 255.255.0.0
Device(config-if)# standby 2 ip 10.0.0.254
Device(config-if)# standby 2 follow HSRP1
! Client group
```

Feature Information for HSRP Support for ICMP Redirects

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Example: Configuring HSRP Virtual MAC Addresses and BIA MAC Address

In an Advanced Peer-to-Peer Networking (APPN) network, an end node is typically configured with the MAC address of the adjacent network node. In the following example, if the end nodes are configured to use 4000.1000.1060, HSRP group 1 is configured to use the same MAC address:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.0.0.1
Device(config-if)# standby 1 mac-address 4000.1000.1060
Device(config-if)# standby 1 ip 10.0.0.11
```

In the following example, the burned-in address of Token Ring interface 3/0 will be the virtual MAC address mapped to the virtual IP address:

```
Device(config)# interface token 3/0
Device(config-if)# standby use-bia
```

Note You cannot use the **standby use-bia** command and the **standby mac-address** command in the same configuration.

Example: Linking IP Redundancy Clients to HSRP Groups

The following example shows HSRP support for a static Network Address Translation (NAT) configuration. The NAT client application is linked to HSRP via the correlation between the name specified by the **standby name** command. Two devices are acting as HSRP active and standby, and the NAT inside interfaces are HSRP enabled and configured to belong to the group named “group1.”

Active Device Configuration

```
Device(config)# interface BVI 10
Device(config-if)# ip address 192.168.5.54 255.255.255.255.0
Device(config-if)# no ip redirects
Device(config-if)# ip nat inside
```

```
Device(config-if)# standby 10 ip 192.168.5.30
Device(config-if)# standby 10 priority 110
Device(config-if)# standby 10 preempt
Device(config-if)# standby 10 name group1
Device(config-if)# standby 10 track Ethernet 2/1
!
!
Device(config)# ip default-gateway 10.0.18.126
Device(config)# ip nat inside source static 192.168.5.33 10.10.10.5 redundancy group1
Device(config)# ip classless
Device(config)# ip route 10.10.10.0 255.255.255.0 Ethernet 2/1
Device(config)# ip route 172.22.33.0 255.255.255.0 Ethernet 2/1
Device(config)# no ip http server
```

Standby Device Configuration

```
Device(config)# interface BVI 10
Device(config-if)# ip address 192.168.5.56 255.255.255.255.0
Device(config-if)# no ip redirects
Device(config-if)# ip nat inside
Device(config-if)# standby 10 priority 95
Device(config-if)# standby 10 preempt
Device(config-if)# standby 10 name group1
Device(config-if)# standby 10 ip 192.168.5.30
Device(config-if)# standby 10 track Ethernet 3/1
Device(config-if)# exit
Device(config)# ip default-gateway 10.0.18.126
Device(config)# ip nat inside source static 192.168.5.33 3.3.3.5 redundancy group1
Device(config)# ip classless
Device(config)# ip route 10.0.32.231 255.255.255 Ethernet 3/1
Device(config)# ip route 10.10.10.0 255.255.255.0 Ethernet 3/1
Device(config)# no ip http server
```

Example: Configuring HSRP Version 2

The following example shows how to configure HSRP version 2 on an interface with a group number of 350:

```
Device(config)# interface vlan 350
Device(config-if)# standby version 2
Device(config-if)# standby 350 priority 110
Device(config-if)# standby 350 preempt
Device(config-if)# standby 350 timers 5 15
Device(config-if)# standby 350 ip 172.20.100.10
```

Example: Enabling SSO-Aware HSRP

The following example shows how to set the redundancy mode to SSO. HSRP is automatically SSO-aware when this mode is enabled.

```
Device(config)# redundancy
Device(config-red)# mode sso
```

If SSO HSRP is disabled using the **no standby sso** command, you can reenable it as shown in the following example:

```
Device(config)# interface GigabitEthernet 1/0/0
```

```
Device(config-if)# ip address 10.1.1.1 255.255.0.0
Device(config-if)# standby priority 200
Device(config-if)# standby preempt
Device(config-if)# standby sso
```

Example: Enabling HSRP MIB Traps

The following examples show how to configure HSRP on two devices and enable the HSRP MIB trap support functionality. As in many environments, one device is preferred as the active one. To configure a device's preference as the active device, configure the device at a higher priority level and enable preemption. In the following example, the active device is referred to as the primary device. The second device is referred to as the backup device:

Device A

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.1.1.1 255.255.0.0
Device(config-if)# standby priority 200
Device(config-if)# standby preempt
Device(config-if)# standby ip 10.1.1.3
Device(config-if)# exit
Device(config)# snmp-server enable traps hsrp
Device(config)# snmp-server host yourhost.cisco.com public hsrp
```

Device B

```
Device(config)#interface GigabitEthernet 1/0/0
Device(config-if)# ip address 10.1.1.2 255.255.0.0
Device(config-if)#  standby priority 101
Device(config-if)# standby ip 10.1.1.3
Device(config-if)# exit
Device(config)# snmp-server enable traps hsrp
Device(config)# snmp-server host myhost.cisco.com public hsrp
```

Example: HSRP BFD Peering

Hot Standby Router Protocol (HSRP) supports Bidirectional Forwarding Detection (BFD) as a part of the HSRP group member health monitoring system. Without BFD, HSRP runs as a process in a multiprocess system and cannot be guaranteed to be scheduled in time to service large numbers of groups with millisecond hello and hold timers. BFD runs as a pseudo-preemptive process and can therefore, be guaranteed to run when required. Only one BFD session between two devices can provide early failover notification for multiple HSRP groups.

In the following example, the **standby bfd** and the **standby bfd all-interfaces** commands are not displayed. HSRP support for BFD is enabled by default when BFD is configured on a device or an interface by using the **bfd interval** command. The **standby bfd** and **standby bfd all-interfaces** commands are needed only if BFD has been manually disabled on a device or an interface.

Device A

```
DeviceA(config)# ip cef
DeviceA(config)# interface FastEthernet2/0
DeviceA(config-if)#  no shutdown
```

```
DeviceA(config-if)# ip address 10.0.0.2 255.0.0.0
DeviceA(config-if)# ip router-cache cef
DeviceA(config-if)# bfd interval 200 min_rx 200 multiplier 3
DeviceA(config-if)# standby 1 ip 10.0.0.11
DeviceA(config-if)# standby 1 preempt
DeviceA(config-if)# standby 1 priority 110
DeviceA(config-if)# standby 2 ip 10.0.0.12
DeviceA(config-if)# standby 2 preempt
DeviceA(config-if)# standby 2 priority 110
```

Device B

```
DeviceB(config)# interface FastEthernet2/0
DeviceB(config-if)# ip address 10.1.0.22 255.255.0.0
DeviceB(config-if)# no shutdown
DeviceB(config-if)# bfd interval 200 min_rx 200 multiplier 3
DeviceB(config-if)# standby 1 ip 10.0.0.11
DeviceB(config-if)# standby 1 preempt
DeviceB(config-if)# standby 1 priority 90
DeviceB(config-if)# standby 2 ip 10.0.0.12
DeviceB(config-if)# standby 2 preempt
DeviceB(config-if)# standby 2 priority 80
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	“HSRP for IPv6” module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HSRP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 64: Feature Information for HSRP

Feature Name	Releases	Feature Information
HSRP	Cisco IOS XE 17.3.2	This feature was introduced for the Catalyst 8000 Series platforms.

Glossary

ARP—Address Resolution Protocol (ARP). ARP performs a required function in IP routing. ARP finds the hardware address, also known as Media Access Control (MAC) address, of a host from its known IP address. ARP maintains a cache (table) in which MAC addresses are mapped to IP addresses. ARP is part of all Cisco IOS systems running IP.

active device—The primary device in an HSRP group that is currently forwarding packets for the virtual device.

active RP—The active RP that controls the system, provides network services, runs the routing protocols, and presents the system management interface.

client group—An HSRP group that is created on a subinterface and linked to the primary group via the group name.

HSRP—Hot Standby Router Protocol. Protocol that provides high network availability and transparent network-topology changes. HSRP creates a router group with a lead device that services all packets sent to the HSRP address. The lead device is monitored by other devices in the group, and if it fails, one of these standby HSRP devices inherits the lead position and the HSRP group address.

ISSU—In Service Software Upgrade. A process that allows Cisco IOS software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco IOS software to be modified while packet forwarding continues, which increases network availability and reduces downtime caused by planned software upgrades.

primary group—An HSRP group that is required on a physical interface for the purposes of electing active and standby devices.

RF—Redundancy Facility. A structured, functional interface used to notify its clients of active and standby state progressions and events.

RP—Route Processor. A generic term for the centralized control unit in a chassis.

RPR—Route Processor Redundancy. RPR provides an alternative to the High System Availability (HSA) feature. HSA enables a system to reset and use a standby Route Processor (RP) if the active RP fails. Using RPR, you can reduce unplanned downtime because RPR enables a quicker switchover between an active and standby RP if the active RP experiences a fatal error.

RPR+—An enhancement to RPR in which the standby RP is fully initialized.

standby group—The set of devices participating in HSRP that jointly emulate a virtual device.

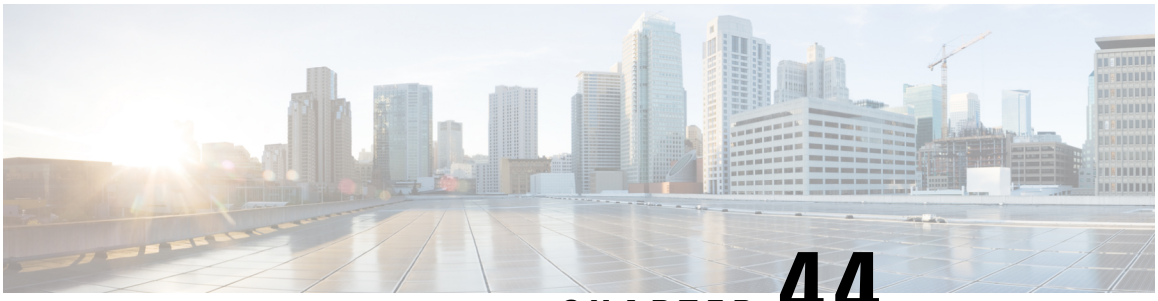
standby device—The backup device in an HSRP group.

standby RP—The backup RP.

switchover—An event in which system control and routing protocol execution are transferred from the active RP to the standby RP. Switchover may be a manual operation or may be induced by a hardware or software fault. Switchover may include transfer of the packet forwarding function in systems that combine system control and packet forwarding in an indivisible unit.

virtual IP address—The default gateway IP address configured for an HSRP group.

virtual MAC address—For Ethernet and FDDI, the automatically generated MAC address when HSRP is configured. The standard virtual MAC address used is: 0000.0C07.AC*xy*, where *xy* is the group number in hexadecimal. The functional address is used for Token Ring. The virtual MAC address is different for HSRP version 2.



CHAPTER 44

HSRP Version 2

Information About HSRP Version 2

HSRP Version 2 Design

HSRP version 2 is designed to address the following restrictions in HSRP version 1:

- In HSRP version 1, millisecond timer values are not advertised or learned. HSRP version 2 advertises and learns millisecond timer values. This change ensures stability of the HSRP groups in all cases.
- In HSRP version 1, group numbers are restricted to the range from 0 to 255. HSRP version 2 expands the group number range from 0 to 4095.
- HSRP version 2 provides improved management and troubleshooting. With HSRP version 1, you cannot use HSRP active hello messages to identify which physical device sent the message because the source MAC address is the HSRP virtual MAC address. The HSRP version 2 packet format includes a 6-byte identifier field that is used to uniquely identify the sender of the message. Typically, this field is populated with the interface MAC address.
- The multicast address 224.0.0.2 is used to send HSRP hello messages. This address can conflict with Cisco Group Management Protocol (CGMP) leave processing.

Version 1 is the default version of HSRP.

HSRP version 2 uses the new IP multicast address 224.0.0.102 to send hello packets instead of the multicast address of 224.0.0.2, used by HSRP version 1. This new multicast address allows CGMP leave processing to be enabled at the same time as HSRP.

HSRP version 2 permits an expanded group number range, 0 to 4095, and consequently uses a new MAC address range 0000.0C9F.F000 to 0000.0C9F.FFFF. The increased group number range does not imply that an interface can, or should, support that many HSRP groups. The expanded group number range was changed to allow the group number to match the VLAN number on subinterfaces.

When the HSRP version is changed, each group will reinitialize because it now has a new virtual MAC address.

HSRP version 2 has a different packet format than HSRP version 1. The packet format uses a type-length-value (TLV) format. HSRP version 2 packets received by an HSRP version 1 device will have the type field mapped to the version field by HSRP version 1 and subsequently ignored.

The Gateway Load Balancing Protocol (GLBP) also addresses the same restrictions relative to HSRP version 1 that HSRP version 2 does. See the *Configuring GLBP* document for more information on GLBP.

Jitter timers

Jitter timers are used in HSRP. They are recommended for timers running on services that work realtime and scale. Jitter timers are intended to significantly improve the reliability of HSRP, and other FHRP protocols, by reducing the chance of bunching of HSRP groups operations, and thus help reduce CPU and network traffic spikes. In the case of HSRP, a given device may have up to 4000 operational groups configured. In order to distribute the load on the device and network, the HSRP timers use a jitter. A given timer instance may take up to 20% more than the configured value. For example, for a hold time set to 15 seconds, the actual hold time may take 18 seconds.

In HSRP, the Hello timer (which sends the Hello Packet) has a negative Jitter, while the Holddown timer (which checks for failure of a peer) has a positive jitter.

How to Configure HSRP Version 2

Changing to HSRP Version 2

HSRP version 2 was introduced to prepare for further enhancements and to expand the capabilities beyond what is possible with HSRP version 1. HSRP version 2 has a different packet format than HSRP version 1.

Note

- HSRP version 2 is not available for ATM interfaces running LAN emulation.
- HSRP version 2 will not interoperate with HSRP version 1. An interface cannot operate both version 1 and version 2 because both versions are mutually exclusive. However, the different versions can be run on different physical interfaces of the same device. You cannot change from version 2 to version 1 if you have configured groups above the group number range allowed for version 1 (0 to 255).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask*
5. **standby version** {**1** | **2**}
6. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
7. **end**
8. **show standby**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface vlan 400`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.10.28.1 255.255.255.0`	Sets an IP address for an interface.
Step 5	**standby version** {**1** | **2**} **Example:** `Device(config-if)# standby version 2`	Changes the HSRP version. **Note** The **standby version 2** command appears in the running configuration as it's a non-default setting. However, the **standby version 1** command doesn't appear because it represents the default HSRP version.
Step 6	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 400 ip 10.10.28.5`	Activates HSRP. • The group number range for HSRP version 2 is 0 through 4095. The group number range for HSRP version 1 is 0 through 255.
Step 7	**end** **Example:** `Device(config-if)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 8	**show standby** **Example:** `Device# show standby`	(Optional) Displays HSRP information. • HSRP version 2 information will be displayed if configured.

Configuration Examples for HSRP Version 2

Example: Configuring HSRP Version 2

The following example shows how to configure HSRP version 2 on an interface with a group number of 350:

```
Device(config)# interface vlan 350
Device(config-if)# standby version 2
Device(config-if)# standby 350 priority 110
Device(config-if)# standby 350 preempt
Device(config-if)# standby 350 timers 5 15
Device(config-if)# standby 350 ip 172.20.100.10
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	"HSRP for IPv6" module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

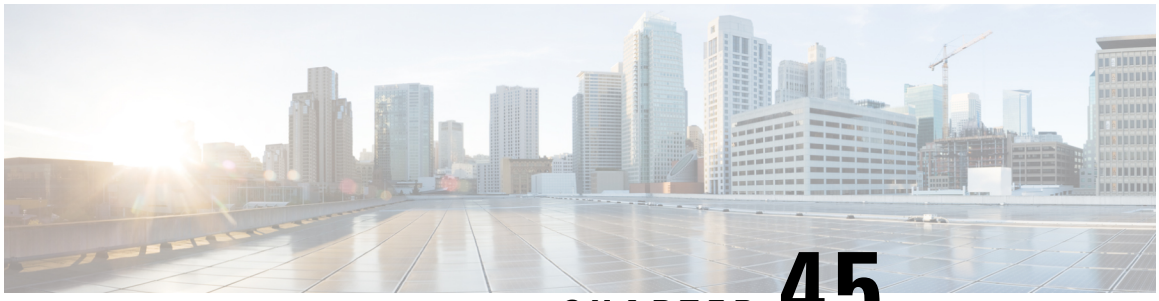
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HSRP Version 2

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 45

HSRP MD5 Authentication

Information About HSRP MD5 Authentication

HSRP Text Authentication

HSRP ignores unauthenticated HSRP protocol messages. The default authentication type is text authentication.

HSRP authentication protects against false HSRP hello packets causing a denial-of-service attack. For example, Device A has a priority of 120 and is the active device. If a host sends spoof HSRP hello packets with a priority of 130, then Device A stops being the active device. If Device A has authentication configured such that the spoof HSRP hello packets are ignored, Device A will remain the active device

HSRP packets will be rejected in any of the following cases:

- The authentication schemes differ on the device and in the incoming packets.
- Text authentication strings differ on the device and in the incoming packet.

HSRP MD5 Authentication

Before the introduction of HSRP MD5 authentication, HSRP authenticated protocol packets with a simple plain text string. HSRP MD5 authentication is an enhancement to generate an MD5 digest for the HSRP portion of the multicast HSRP protocol packet. This functionality provides added security and protects against the threat from HSRP-spoofing software.

MD5 authentication provides greater security than the alternative plain text authentication scheme. MD5 authentication allows each HSRP group member to use a secret key to generate a keyed MD5 hash that is part of the outgoing packet. A keyed hash of an incoming packet is generated and if the hash within the incoming packet does not match the generated hash, the packet is ignored.

The key for the MD5 hash can be either given directly in the configuration using a key string or supplied indirectly through a key chain.

HSRP has two authentication schemes:

- Plain text authentication
- MD5 authentication

HSRP authentication protects against false HSRP hello packets causing a denial-of-service attack. For example, Device A has a priority of 120 and is the active device. If a host sends spoof HSRP hello packets with a priority of 130, then Device A stops being the active device. If Device A has authentication configured such that the spoof HSRP hello packets are ignored, Device A will remain the active device.

HSRP packets will be rejected in any of the following cases:

- The authentication schemes differ on the device and in the incoming packets.
- MD5 digests differ on the device and in the incoming packet.
- Text authentication strings differ on the device and in the incoming packet.

How to Configure HSRP MD5 Authentication

Configuring HSRP MD5 Authentication Using a Key Chain

Perform this task to configure HSRP MD5 authentication using a key chain. Key chains allow a different key string to be used at different times according to the key chain configuration. HSRP will query the appropriate key chain to obtain the current live key and key ID for the specified key chain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. **key** *key-id*
5. **key-string** *string*
6. **exit**
7. **exit**
8. **interface** *type number*
9. **ip address** *ip-address mask* [**secondary**]
10. **standby** [*group-number*] **priority** *priority*
11. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*]
12. **standby** [*group-number*] **authentication md5 key-chain** *key-chain-name*
13. **standby** [*group-number*] **ip** [*ip-address* [*secondary*]]
14. Repeat Steps 1 through 12 on each device that will communicate.
15. **end**
16. **show standby**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain hsrp1`	Enables authentication for routing protocols, identifies a group of authentication keys, and enters key-chain configuration mode.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 100`	Identifies an authentication key on a key chain and enters key-chain key configuration mode. • The value for the*key-id* argument must be a number.
Step 5	**key-string** *string* **Example:** `Device(config-keychain-key)# key-string mno172`	Specifies the authentication string for a key. • The value for the *string* argument can be 1 to 80 uppercase or lowercase alphanumeric characters; the first character cannot be a numeral
Step 6	**exit** **Example:** `Device(config-keychain-key)# exit`	Returns to key-chain configuration mode.
Step 7	**exit** **Example:** `Device(config-keychain)# exit`	Returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.21.8.32 255.255.255.0`	Specifies a primary or secondary IP address for an interface.

	Command or Action	Purpose
Step 10	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 11	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption.
Step 12	**standby** [*group-number*] **authentication md5 key-chain** *key-chain-name* **Example:** `Device(config-if)# standby 1 authentication md5 key-chain hsrp1`	Configures an authentication MD5 key chain for HSRP MD5 authentication. • The key chain name must match the name specified in Step 3.
Step 13	**standby** [*group-number*] **ip** [*ip-address* [*secondary*]] **Example:** `Device(config-if)# standby 1 ip 10.21.8.12`	Activates HSRP.
Step 14	Repeat Steps 1 through 12 on each device that will communicate.	—
Step 15	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 16	**show standby** **Example:** `Device# show standby`	(Optional) Displays HSRP information. • Use this command to verify your configuration. The key string or key chain will be displayed if configured.

Troubleshooting HSRP MD5 Authentication

Perform this task if HSRP MD5 authentication is not operating correctly.

SUMMARY STEPS

1. **enable**
2. **debug standby errors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug standby errors** **Example:** `Device# debug standby errors`	Displays error messages related to HSRP. • Error messages will be displayed for each packet that fails to authenticate, so use this command with care.

Examples

In the following example, Device A has MD5 text string authentication configured, but Device B has the default text authentication:

```
Device# debug standby errors

A:Jun 16 12:14:50.337:HSRP:Et0/1 Grp 0 Auth failed for Hello pkt from 10.21.0.5, MD5 confgd
 but no tlv
B:Jun 16 12:16:34.287:HSRP:Et0/1 Grp 0 Auth failed for Hello pkt from 10.21.0.4, Text auth
 failed
```

In the following example, both Device A and Device B have different MD5 authentication strings:

```
Device# debug standby errors

A:Jun 16 12:19:26.335:HSRP:Et0/1 Grp 0 Auth failed for Hello pkt from 10.21.0.5, MD5 auth
failed
B:Jun 16 12:18:46.280:HSRP:Et0/1 Grp 0 Auth failed for Hello pkt from 10.21.0.4, MD5 auth
failed
```

Configuring HSRP Text Authentication

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby** [*group-number*] **priority** *priority*
6. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*]
7. **standby** [*group-number*] **authentication text** *string*
8. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
9. Repeat Steps 1 through 8 on each device that will communicate.
10. **end**

11. **show standby**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 6	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption.
Step 7	**standby** [*group-number*] **authentication text** *string* **Example:** `Device(config-if)# standby 1 authentication text authentication1`	Configures an authentication string for HSRP text authentication. • The default string is cisco.
Step 8	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3`	Activates HSRP.

	Command or Action	Purpose
Step 9	Repeat Steps 1 through 8 on each device that will communicate.	--
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 11	**show standby** **Example:** `Device# show standby`	(Optional) Displays HSRP information. • Use this command to verify your configuration. The key string or key chain will be displayed if configured.

Configuration Examples for HSRP MD5 Authentication

Example: Configuring HSRP MD5 Authentication Using Key Strings

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 authentication md5 key-string 54321098452103ab timeout 30
Device(config-if)# standby 1 ip 10.21.0.10
```

Example: Configuring HSRP MD5 Authentication Using Key Chains

In the following example, HSRP queries the key chain “hsrp1” to obtain the current live key and key ID for the specified key chain:

```
Device(config)# key chain hsrp1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 54321098452103ab
Device(config-keychain-key)# exit
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 authentication md5 key-chain hsrp1
Device(config-if)# standby 1 ip 10.21.0.10
```

Example: Configuring HSRP MD5 Authentication Using Key Strings and Key Chains

The key ID for key-string authentication is always zero. If a key chain is configured with a key ID of zero, then the following configuration will work:

Device 1

```
Device(config)# key chain hsrp1
Device(config-keychain)# key 0
Device(config-keychain-key)# key-string 54321098452103ab
Device(config-keychain-key)# exit
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 authentication md5 key-chain hsrp1
Device(config-if)# standby 1 ip 10.21.0.10
```

Device 2

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 authentication md5 key-string 54321098452103ab
Device(config-if)# standby 1 ip 10.21.0.10
```

Example: Configuring HSRP Text Authentication

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 authentication text company2
Device(config-if)# standby 1 ip 10.21.0.10
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	"HSRP for IPv6" module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

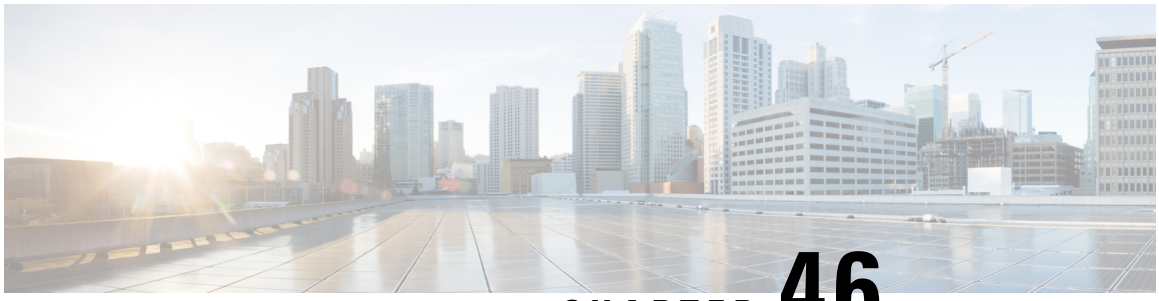
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HSRP MD5 Authentication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 46

HSRP Support for ICMP Redirects

Information About HSRP Support for ICMP Redirects

HSRP Support for ICMP Redirect Messages

By default, HSRP filtering of Internet Control Message Protocol (ICMP) redirect messages is enabled on devices running HSRP.

ICMP is a network layer Internet protocol that provides message packets to report errors and other information relevant to IP processing. ICMP can send error packets to a host and can send redirect packets to a host.

When HSRP is running, preventing hosts from discovering the interface (or real) IP addresses of devices in the HSRP group is important. If a host is redirected by ICMP to the real IP address of a device, and that device later fails, then packets from the host will be lost.

ICMP redirect messages are automatically enabled on interfaces configured with HSRP. This functionality works by filtering outgoing ICMP redirect messages through HSRP, where the next hop IP address may be changed to an HSRP virtual IP address.

ICMP Redirects to Active HSRP Devices

The next-hop IP address is compared to the list of active HSRP devices on that network; if a match is found, then the real next-hop IP address is replaced with a corresponding virtual IP address and the redirect message is allowed to continue.

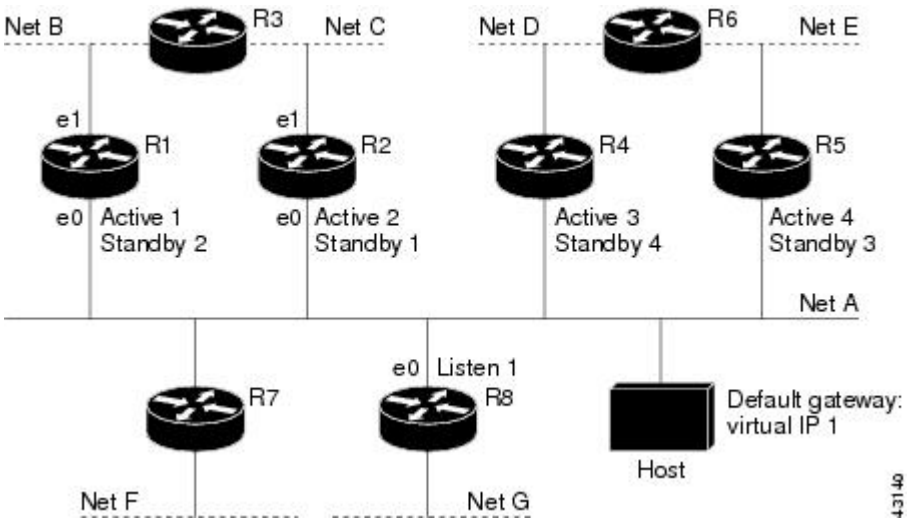
If no match is found, then the ICMP redirect message is sent only if the device corresponding to the new next hop IP address is not running HSRP. Redirects to passive HSRP devices are not allowed (a passive HSRP device is a device running HSRP, but which contains no active HSRP groups on the interface).

For optimal operation, every device in a network that is running HSRP should contain at least one active HSRP group on an interface to that network. Every HSRP device need not be a member of the same group.

Each HSRP device will snoop on all HSRP packets on the network to maintain a list of active devices (virtual IP addresses versus real IP addresses).

Consider the network shown in the figure below, which supports the HSRP ICMP redirection filter.

Figure 14: Network Supporting the HSRP ICMP Redirection Filter



If the host wants to send a packet to another host on Net D, then it first sends it to its default gateway, the virtual IP address of HSRP group 1.

The following is the packet received from the host:

```
dest MAC         = HSRP group 1 virtual MAC
source MAC       = Host MAC
dest IP          = host-on-netD IP
source IP        = Host IP
```

Device R1 receives this packet and determines that device R4 can provide a better path to Net D, so it prepares to send a redirect message that will redirect the host to the real IP address of device R4 (because only real IP addresses are in its routing table).

The following is the initial ICMP redirect message sent by device R1:

```
dest MAC         = Host MAC
source MAC       = router R1 MAC
dest IP          = Host IP
source IP        = router R1 IP
gateway to use   = router R4 IP
```

Before this redirect occurs, the HSRP process of device R1 determines that device R4 is the active HSRP device for group 3, so it changes the next hop in the redirect message from the real IP address of device R4 to the virtual IP address of group 3. Furthermore, it determines from the destination MAC address of the packet that triggered the redirect message that the host used the virtual IP address of group 1 as its gateway, so it changes the source IP address of the redirect message to the virtual IP address of group 1.

The modified ICMP redirect message showing the two modified fields (*) is as follows:

```
dest MAC         = Host MAC
source MAC       = router R1 MAC
dest IP          = Host IP
```

```
source IP*       = HSRP group 1 virtual IP
gateway to use*  = HSRP group 3 virtual IP
```

This second modification is necessary because hosts compare the source IP address of the ICMP redirect message with their default gateway. If these addresses do not match, the ICMP redirect message is ignored. The routing table of the host now consists of the default gateway, virtual IP address of group 1, and a route to Net D through the virtual IP address of group 3.

ICMP Redirects to Passive HSRP Devices

ICMP redirects to passive HSRP devices are not permitted. Redundancy may be lost if hosts learn the real IP addresses of HSRP devices.

In the "Network Supporting the HSRP ICMP Redirection Filter" figure, redirection to device R8 is not allowed because R8 is a passive HSRP device. In this case, packets from the host to Net D will first go to device R1 and then be forwarded to device R4; that is, they will traverse the network twice.

A network configuration with passive HSRP devices is considered a misconfiguration. For HSRP ICMP redirection to operate optimally, every device on the network that is running HSRP should contain at least one active HSRP group.

ICMP Redirects to Non-HSRP Devices

ICMP redirects to devices not running HSRP on their local interface are permitted. No redundancy is lost if hosts learn the real IP address of non-HSRP devices.

In the "Network Supporting the HSRP ICMP Redirection Filter" figure, redirection to device R7 is allowed because R7 is not running HSRP. In this case, the next hop IP address is unchanged. The source IP address is changed dependent upon the destination MAC address of the original packet. You can specify the **no standby redirect unknown** command to stop these redirects from being sent.

Passive HSRP Advertisement Messages

Passive HSRP devices send out HSRP advertisement messages both periodically and when entering or leaving the passive state. Thus, all HSRP devices can determine the HSRP group state of any HSRP device on the network. These advertisements inform other HSRP devices on the network of the HSRP interface state, as follows:

- Active—Interface has at least one active group. A single advertisement is sent out when the first group becomes active.
- Dormant—Interface has no HSRP groups. A single advertisement is sent once when the last group is removed.
- Passive—Interface has at least one nonactive group and no active groups. Advertisements are sent out periodically.

You can adjust the advertisement interval and hold-down time using the **standby redirect timers** command.

ICMP Redirects Not Sent

If the HSRP device cannot uniquely determine the IP address used by the host when it sends the packet that caused the redirect, the redirect message will not be sent. The device uses the destination MAC address in the

original packet to make this determination. In certain configurations, such as the use of the **standby use-bia** interface configuration command specified on an interface, redirects cannot be sent. In this case, the HSRP groups use the interface MAC address as their virtual MAC address. The device now cannot determine if the default gateway of the host is the real IP address or one of the HSRP virtual IP addresses that are active on the interface.

The IP source address of an ICMP packet must match the gateway address used by the host in the packet that triggered the ICMP packet, otherwise the host will reject the ICMP redirect packet. An HSRP device uses the destination MAC address to determine the gateway IP address of the host. If the HSRP device is using the same MAC address for multiple IP addresses, uniquely determining the gateway IP address of the host is not possible, and the redirect message is not sent.

The following is sample output from the **debug standby events icmp** EXEC command if HSRP could not uniquely determine the gateway used by the host:

```
10:43:08: HSRP: ICMP redirect not sent to 10.0.0.4 for dest 10.0.1.2
10:43:08: HSRP: could not uniquely determine IP address for mac 00d0.bbd3.bc22
```

How to Configure HSRP Support for ICMP Redirects

Enabling HSRP Support for ICMP Redirect Messages

By default, HSRP filtering of ICMP redirect messages is enabled on devices running HSRP. Perform this task to reenable this feature on your device if it is disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **standby redirect** [**timers** *advertisement holddown*] [**unknown**]
5. **end**
6. **show standby redirect** [*ip-address*] [*interface-type interface-number*] [**active**] [**passive**] [**timers**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**standby redirect** [**timers** *advertisement holddown*] [**unknown**] **Example:** `Device(config-if)# standby redirect`	Enables HSRP filtering of ICMP redirect messages. • You can also use this command in global configuration mode, which enables HSRP filtering of ICMP redirect messages on all interfaces configured for HSRP.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show standby redirect** [*ip-address*] [*interface-type interface-number*] [**active**] [**passive**] [**timers**] **Example:** `Device# show standby redirect`	(Optional) Displays ICMP redirect information on interfaces configured with HSRP.

Configuration Examples for HSRP Support for ICMP Redirects

Example: Configuring HSRP Support for ICMP Redirect Messages

Device A Configuration—Active for Group 1 and Standby for Group 2

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.0.0.10 255.0.0.0
Device(config-if)# standby redirect
Device(config-if)# standby 1 priority 120
Device(config-if)# standby 1 preempt delay minimum 20
Device(config-if)# standby 1 ip 10.0.0.1
Device(config-if)# standby 2 priority 105
Device(config-if)# standby 2 preempt delay minimum 20
Device(config-if)# standby 2 ip 10.0.0.2
```

Device B Configuration—Standby for Group 1 and Active for Group 2

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.0.0.11 255.0.0.0
Device(config-if)# standby redirect
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 preempt delay minimum 20
Device(config-if)# standby 1 ip 10.0.0.1
```

```
Device(config-if)# standby 2 priority 120
Device(config-if)# standby 2 preempt delay minimum 20
Device(config-if)# standby 2 ip 10.0.0.2
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	"HSRP for IPv6" module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

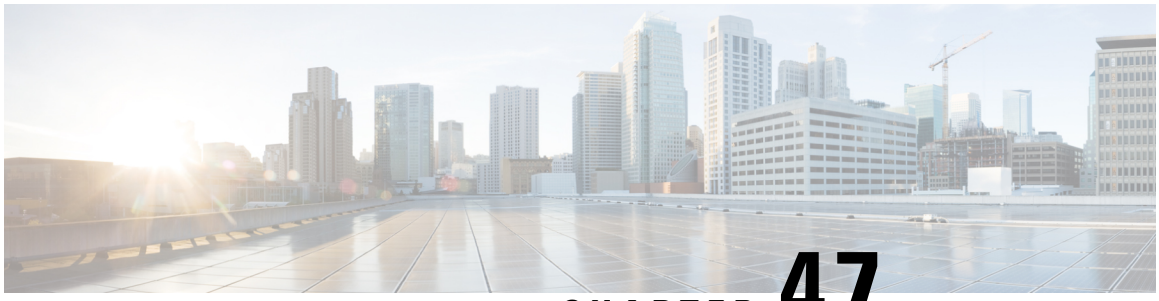
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HSRP Support for ICMP Redirects

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 47

FHRP - HSRP Multiple Group Optimization

Information About FHRP - Multiple Group Optimization

HSRP Multiple Group Optimization

The configuration of many hundreds of subinterfaces on the same physical interface, with each subinterface having its own HSRP group, can cause the processes of negotiation and maintenance of multiple HSRP groups to have a detrimental impact on network traffic and CPU utilization.

Only one HSRP group is required on a physical interface for the purposes of electing active and standby devices. This group is known as the *primary* group. Other HSRP groups may be created on each subinterface and linked to the primary group via the group name. These linked HSRP groups are known as *client* or *subordinate* groups.

The HSRP group state of the client groups follows that of the primary group. Client groups do not participate in any sort of device election mechanism.

Client groups send periodic messages in order to refresh their virtual MAC addresses in switches and learning bridges. The refresh message may be sent at a much lower frequency compared with the protocol election messages sent by the primary group.

How to configure FHRP - Multiple Group Optimization

Configuring Multiple HSRP Groups for Load Balancing

Perform this task to configure multiple HSRP groups for load balancing.

Multiple HSRP groups enable redundancy and load-sharing within networks and allow redundant devices to be more fully utilized. A device actively forwarding traffic for one HSRP group can be in standby or in the listen state for another group.

If two devices are used, then Device A would be configured as active for group 1 and standby for group 2. Device B would be standby for group 1 and active for group 2. Fifty percent of the hosts on the LAN would be configured with the virtual IP address of group 1 and the remaining hosts would be configured with the virtual IP address of group 2. See the Example: Configuring Multiple HSRP Groups for Load Balancing for a diagram and configuration example.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby** [*group-number*] **priority** *priority*
6. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *delay*]
7. **standby** [*group-number*] **ip** [*ip-address*] **secondary**]
8. On the same device, repeat Steps 5 through 7 to configure the device attributes for different standby groups.
9. **exit**
10. Repeat Steps 3 through 9 on another device.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.

	Command or Action	Purpose
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 6	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *delay*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption.
Step 7	**standby** [*group-number*] **ip** [*ip-address*] **secondary**] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3`	Activates HSRP.
Step 8	On the same device, repeat Steps 5 through 7 to configure the device attributes for different standby groups.	For example, Device A can be configured as an active device for group 1 and be configured as an active or standby device for another HSRP group with different priority and preemption values.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 10	Repeat Steps 3 through 9 on another device.	Configures multiple HSRP and enables load balancing on another device.

Improving CPU and Network Performance with HSRP Multiple Group Optimization

Perform this task to configure multiple HSRP client groups.

The **standby follow** command configures an HSRP group to become a subordinate of another HSRP group.

HSRP client groups follow the primary HSRP with a slight, random delay so that all client groups do not change at the same time.

Use the **standby mac-refresh** *seconds* command to directly change the HSRP client group refresh interval. The default interval is 10 seconds and can be configured to as much as 255 seconds.

Note

- Client or subordinate groups must be on the same physical interface as the primary group.
- A client group takes its state from the group it is following. Therefore, the client group does not use its timer, priority, or preemption settings. A warning is displayed if these settings are configured on a client group:

```
Device(config-if)# standby 1 priority 110
%Warning: This setting has no effect while following another group.
Device(config-if)# standby 1 timers 5 15
    % Warning: This setting has no effect while following another group.
Device(config-if)# standby 1 preempt delay minimum 300
    % Warning: This setting has no effect while following another group.
```

Before you begin

Configure the HSRP primary group using the steps in the Configuring Multiple HSRP Groups for Load Balancing section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby mac-refresh** *seconds*
6. **standby** *group-number* **follow** *group-name*
7. **exit**
8. Repeat Steps 3 through 6 to configure additional HSRP client groups.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.

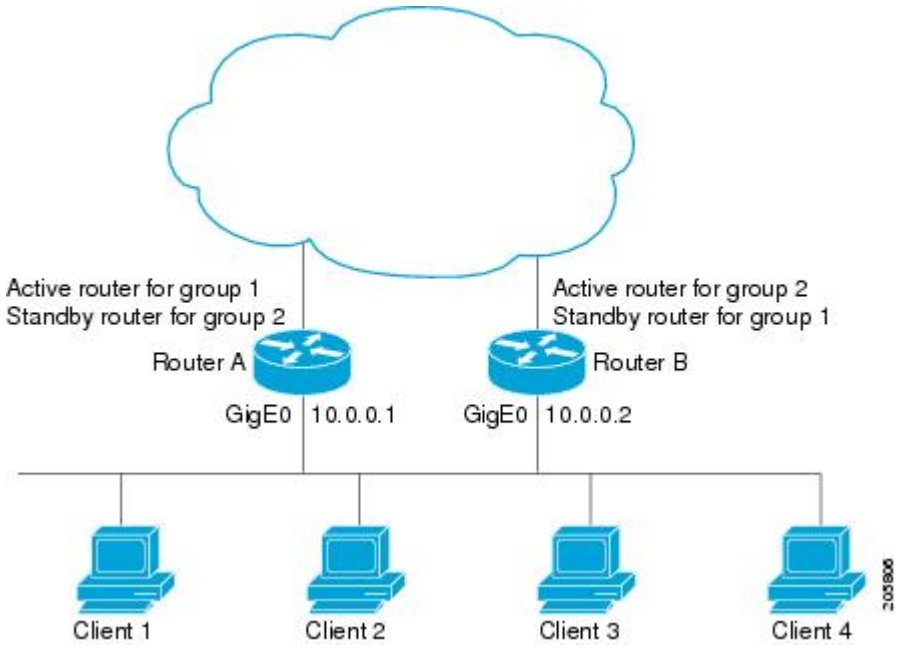
	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby mac-refresh** *seconds* **Example:** `Device(config-if)# standby mac-refresh 30`	Configures the HSRP client group refresh interval.
Step 6	**standby** *group-number* **follow** *group-name* **Example:** `Device(config-if)# standby 1 follow HSRP1`	Configures an HSRP group as a client group.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	Repeat Steps 3 through 6 to configure additional HSRP client groups.	Configures multiple HSRP client groups.

Configuration Examples for FHRP - Multiple Group Optimization

Example: Configuring Multiple HSRP Groups for Load Balancing

You can use HSRP or multiple HSRP groups when you configure load sharing. In the figure below, half of the clients are configured for Router A, and half of the clients are configured for Router B. Together, the configuration for Routers A and B establish two Hot Standby groups. For group 1, Router A is the default active router because it has the assigned highest priority, and Router B is the standby router. For group 2, Router B is the default active router because it has the assigned highest priority, and Router A is the standby router. During normal operation, the two routers share the IP traffic load. When either router becomes unavailable, the other router becomes active and assumes the packet-transfer functions of the router that is unavailable. The **standby preempt** interface configuration command is necessary so that if a router goes down and then comes back up, preemption occurs and restores load sharing.

Figure 15: HSRP Load Sharing Example



The following example shows Router A configured as the active router for group 1 with a priority of 110 and Router B configured as the active router for group 2 with a priority of 110. The default priority level is 100. Group 1 uses a virtual IP address of 10.0.0.3 and Group 2 uses a virtual IP address of 10.0.0.4.

Router A Configuration

```
Router(config)# hostname RouterA
!
Router(config)# interface GigabitEthernet 0/0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.0
Router(config-if)# standby 1 priority 110
Router(config-if)# standby 1 preempt
Router(config-if)# standby 1 ip 10.0.0.3
Router(config-if)# standby 2 preempt
Router(config-if)# standby 2 ip 10.0.0.4
```

Router B Configuration

```
Router(config)# hostname RouterB
!
Router(config)# interface GigabitEthernet 0/0/0
Router(config-if)# ip address 10.0.0.2 255.255.255.0
Router(config-if)# standby 1 preempt
Router(config-if)# standby 1 ip 10.0.0.3
Router(config-if)# standby 2 priority 110
Router(config-if)# standby 2 preempt
Router(config-if)# standby 2 ip 10.0.0.4
```

Example: Improving CPU and Network Performance with HSRP Multiple Group Optimization

The following example shows how to configure an HSRP client and primary group:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# no shutdown
Device(config-if)# standby mac-refresh 30
! Client Hello message interval
!
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# no shutdown
Device(config-if)# ip vrf forwarding VRF2
Device(config-if)# ip address 10.0.0.100 255.255.0.0
Device(config-if)# standby 1 ip 10.0.0.254
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 name HSRP1
!Server group
!
Device(config)# interface GigabitEthernet 0/0/2
Device(config-if)# no shutdown
Device(config-if)# ip vrf forwarding VRF3
Device(config-if)# ip address 10.0.0.100 255.255.0.0
Device(config-if)# standby 2 ip 10.0.0.254
Device(config-if)# standby 2 follow HSRP1
! Client group
!
Device(config)# interface GigabitEthernet 0/0/3
Device(config-if)# no shutdown
Device(config-if)# ip vrf forwarding VRF4
Device(config-if)# ip address 10.0.0.100 255.255.0.0
Device(config-if)# standby 2 ip 10.0.0.254
Device(config-if)# standby 2 follow HSRP1
! Client group
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	“HSRP for IPv6” module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

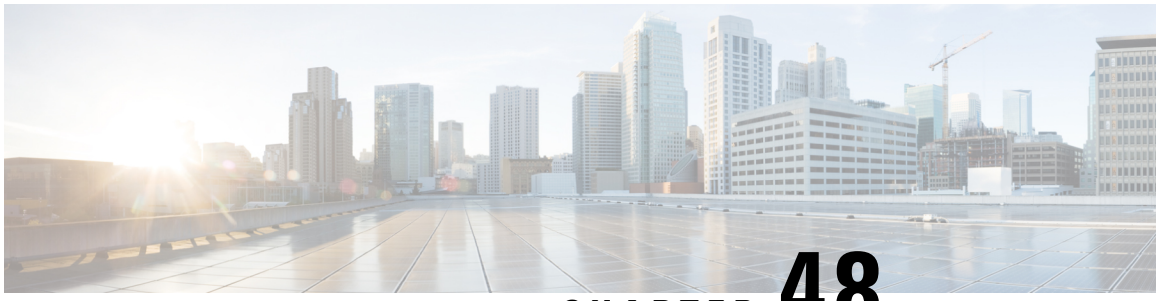
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for FHRP - HSRP Multiple Group Optimization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 48

FHRP - HSRP Group Shutdown

Information About FHRP - HSRP Group Shutdown

How Object Tracking Affects the Priority of an HSRP Device

The priority of a device can change dynamically if it has been configured for object tracking and the object that is being tracked goes down. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to HSRP, either immediately or after a specified delay. The object values are reported as either up or down. Examples of objects that can be tracked are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, the HSRP priority is reduced. The HSRP device with the higher priority can become the active device if it has the **standby preempt** command configured.

HSRP Object Tracking

Object tracking separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by any other process as well as HSRP. The priority of a device can change dynamically when it has been configured for object tracking and the object that is being tracked goes down. Examples of objects that can be tracked are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, the HSRP priority is reduced.

A client process such as HSRP, Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) can register its interest in tracking objects and then be notified when the tracked object changes state.

For more information about object tracking, see the "Configuring Enhanced Object Tracking" document.

HSRP Group Shutdown

The FHRP—HSRP Group Shutdown feature enables you to configure an HSRP group to become disabled (its state changed to Init) instead of having its priority decremented when a tracked object goes down. Use the **standby track** command with the **shutdown** keyword to configure HSRP group shutdown.

If an object is already being tracked by an HSRP group, you cannot change the configuration to use the HSRP Group Shutdown feature. You must first remove the tracking configuration using the **no standby track** command and then reconfigure it using the **standby track** command with the **shutdown** keyword.

How to Configure FHRP - HSRP Group Shutdown

Configuring HSRP Object Tracking

Perform this task to configure HSRP to track an object and change the HSRP priority based on the state of the object.

Each tracked object is identified by a unique number that is specified on the tracking CLI. Client processes use this number to track a specific object.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **interface** *type number* {**line-protocol** | **ip routing**}
4. **exit**
5. **interface** *type number*
6. **standby** [*group-number*] **track** *object-number* [**decrement** *priority-decrement*] [**shutdown**]
7. **standby** [*group-number*] **ip** [*ip-address* [**secondary**]]
8. **end**
9. **show track** [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**track** *object-number* **interface** *type number* {**line-protocol** | **ip routing**} **Example:** `Device(config)# track 100 interface GigabitEthernet 0/0/0 line-protocol`	Configures an interface to be tracked and enters tracking configuration mode.
Step 4	**exit** **Example:** `Device(config-track)# exit`	Returns to global configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 6	**standby** [*group-number*] **track** *object-number* [**decrement** *priority-decrement*] [**shutdown**] **Example:** `Device(config-if)# standby 1 track 100 decrement 20`	Configures HSRP to track an object and change the Hot Standby priority on the basis of the state of the object. • By default, the priority of the device is decreased by 10 if a tracked object goes down. Use the **decrement** *priority-decrement* keyword and argument combination to change the default behavior. • When multiple tracked objects are down and *priority-decrement* values have been configured, these configured priority decrements are cumulative. If tracked objects are down, but none of them were configured with priority decrements, the default decrement is 10 and it is cumulative. • Use the **shutdown** keyword to disable the HRSP group on the device when the tracked object goes down. **Note** If an object is already being tracked by an HSRP group, you cannot change the configuration to use the HSRP Group Shutdown feature. You must first remove the tracking configuration using the **no standby track** command and then reconfigure it using the **standby track** command with the **shutdown** keyword.
Step 7	**standby** [*group-number*] **ip** [*ip-address* [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.10.10.0`	Activates HSRP. • The default group number is 0. The group number range is from 0 to 255 for HSRP version 1 and from 0 to 4095 for HSRP version 2.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 9	**show track** [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**] **Example:** `Device# show track 100 interface`	Displays tracking information.

Configuring HSRP MD5 Authentication Using a Key String

Note Text authentication cannot be combined with MD5 authentication for an HSRP group at any one time. When MD5 authentication is configured, the text authentication field in HSRP hello messages is set to all zeroes on transmit and ignored on receipt, provided the receiving device also has MD5 authentication enabled.

Note If you are changing a key string in a group of devices, change the active device last to prevent any HSRP state change. The active device should have its key string changed no later than one hold-time period, specified by the **standy timers** interface configuration command, after the nonactive devices. This procedure ensures that the nonactive devices do not time out the active device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **terminal interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby** [*group-number*] **priority** *priority*
6. **standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*]
7. **standby** [*group-number*] **authentication md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*]
8. **standby** [*group-number*] **ip** [*ip-address*] [**secondary**]]
9. Repeat Steps 1 through 8 on each device that will communicate.
10. **end**
11. **show standby**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**terminal interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures HSRP priority.
Step 6	**standby** [*group-number*] **preempt** [**delay** {**minimum** | **reload** | **sync**} *seconds*] **Example:** `Device(config-if)# standby 1 preempt`	Configures HSRP preemption.
Step 7	**standby** [*group-number*] **authentication md5 key-string** [**0** | **7**] *key* [**timeout** *seconds*] **Example:** `Device(config-if)# standby 1 authentication md5 key-string d00b4r987654321a timeout 30`	Configures an authentication string for HSRP MD5 authentication. • The *key* argument can be up to 64 characters in length. We recommended that at least 16 characters be used. • No prefix to the *key* argument or specifying **0** means the key will be unencrypted. • Specifying **7** means the key will be encrypted. The key-string authentication key will automatically be encrypted if the **service password-encryption** global configuration command is enabled. • The **timeout** value is the period of time that the old key string will be accepted to allow configuration of all routers in a group with a new key.

	Command or Action	Purpose
Step 8	**standby** [*group-number*] **ip** [*ip-address*] [**secondary**]] **Example:** `Device(config-if)# standby 1 ip 10.0.0.3`	Activates HSRP.
Step 9	Repeat Steps 1 through 8 on each device that will communicate.	—
Step 10	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 11	**show standby** **Example:** `Device# show standby`	(Optional) Displays HSRP information. • Use this command to verify your configuration. The key string or key chain will be displayed if configured.

Configuration Examples for FHRP - HSRP Group Shutdown

Example: Configuring HSRP Object Tracking

In the following example, the tracking process is configured to track the IP-routing capability of serial interface 1/0. HSRP on Gigabit Ethernet interface 0/0/0 then registers with the tracking process to be informed of any changes to the IP-routing state of serial interface 1/0. If the IP state on serial interface 1/0 goes down, the priority of the HSRP group is reduced by 10.

If both serial interfaces are operational, Device A will be the HSRP active device because it has the higher priority. However, if IP routing on serial interface 1/0 in Device A fails, the HSRP group priority will be reduced and Device B will take over as the active device, thus maintaining a default virtual gateway service to hosts on the 10.1.0.0 subnet.

Device A Configuration

```
Device(config)# track 100 interface serial 1/0/0 ip routing
!
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 decrement 10
Device(config-if)# standby 1 ip 10.1.0.1
```

Device B Configuration

```
Device(config)# track 100 interface serial 1/0/0 ip routing
!
```

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.1.0.22 255.255.0.0
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 decrement 10
Device(config-if)# standby 1 ip 10.1.0.1
```

Example: Configuring HSRP Group Shutdown

In the following example, the tracking process is configured to track the IP-routing capability of Gigabit Ethernet interface 0/0/0. HSRP on Gigabit Ethernet interface 0/0/1 then registers with the tracking process to be informed of any changes to the IP-routing state of Gigabit Ethernet interface 0/0/0. If the IP state on Gigabit Ethernet interface 0/0/0 goes down, the HSRP group is disabled.

If both Gigabit Ethernet interfaces are operational, Device A will be the HSRP active device because it has the higher priority. However, if IP routing on Gigabit Ethernet interface 0/0/0 in Device A fails, the HSRP group will be disabled and Device B will take over as the active device, thus maintaining a default virtual gateway service to hosts on the 10.1.0.0 subnet.

Device A Configuration

```
Device(config)# track 100 interface GigabitEthernet 0/0/0 ip routing
!
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip address 10.1.0.21 255.255.0.0
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 110
Device(config-if)# standby 1 track 100 shutdown
```

Device B Configuration

```
Device(config)# track 100 interface GigabitEthernet 0/0/0 ip routing
!
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# ip address 10.1.0.22 255.255.0.0
Device(config-if)# standby 1 ip 10.1.0.1
Device(config-if)# standby 1 preempt
Device(config-if)# standby 1 priority 105
Device(config-if)# standby 1 track 100 shutdown
```

If an object is already being tracked by an HSRP group, you cannot change the configuration to use the HSRP Group Shutdown feature. You must first remove the tracking configuration using the **no standby track** command and then reconfigure it using the **standby track** command with the **shutdown** keyword.

The following example shows how to change the configuration of a tracked object to include the HSRP Group Shutdown feature:

```
Device(config)# no standby 1 track 100 decrement 10
Device(config)# standby 1 track 100 shutdown
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	"HSRP for IPv6" module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

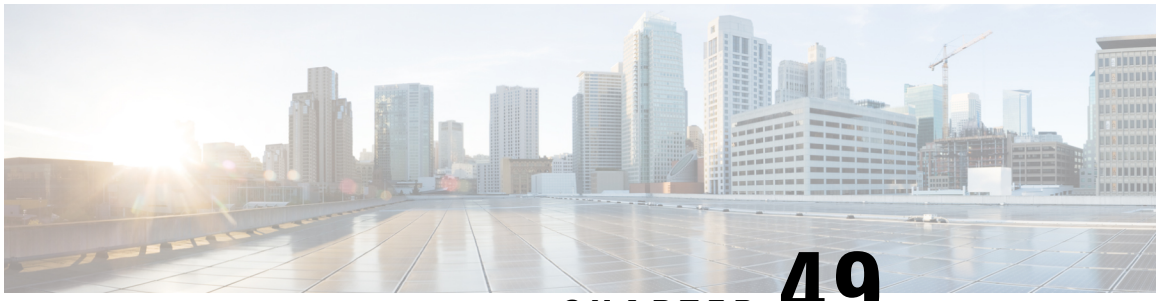
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for FHRP - HSRP Group Shutdown

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 49

SSO HSRP

Restrictions for SSO HSRP

- Enhanced Object Tracking (EOT) is not stateful switchover (SSO)-aware and cannot be used with HSRP in SSO mode.

Information About SSO HSRP

SSO HSRP

SSO HSRP alters the behavior of HSRP when a device with redundant Route Processors (RPs) is configured for stateful switchover (SSO) redundancy mode. When an RP is active and the other RP is standby, SSO enables the standby RP to take over if the active RP fails.

With this functionality, HSRP SSO information is synchronized to the standby RP, allowing traffic that is sent using the HSRP virtual IP address to be continuously forwarded during a switchover without a loss of data or a path change. Additionally, if both RPs fail on the active HSRP device, then the standby HSRP device takes over as the active HSRP device.

The feature is enabled by default when the redundancy mode of operation is set to SSO.

SSO Dual-Route Processors and Cisco Nonstop Forwarding

SSO functions in networking devices (usually edge devices) that support dual RPs. SSO provides RP redundancy by establishing one of the RPs as the active processor and the other RP as the standby processor. SSO also synchronizes critical state information between the RPs so that network state information is dynamically maintained between RPs.

SSO is generally used with Cisco nonstop forwarding (NSF). Cisco NSF enables forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With NSF, users are less likely to experience service outages.

HSRP and SSO Working Together

The SSO HSRP feature enables the Cisco IOS HSRP subsystem software to detect that a standby RP is installed and the system is configured in SSO redundancy mode. Further, if the active RP fails, no change occurs to the HSRP group itself and traffic continues to be forwarded through the current active gateway device.

Prior to introduction of the SSO HSRP feature, when the primary RP of the active device failed, it would stop participating in the HSRP group and trigger another switch in the group to take over as the active HSRP switch.

SSO HSRP is required to preserve the forwarding path for traffic destined to the HSRP virtual IP address through an RP switchover.

Configuring SSO on the edge device enables the traffic on the Ethernet links to continue during an RP failover without the Ethernet traffic switching over to an HSRP standby device (and then back, if preemption is enabled).

Note You may want to disable SSO HSRP by using the **no standby sso** command if you have LAN segments that should switch HSRP traffic to a redundant device while SSO maintains traffic flow for other connections.

How to Configure SSO HSRP

Enabling SSO Aware HSRP

The SSO aware HSRP is enabled by default when the redundancy mode is set to SSO. Perform this task to reenable HSRP to be SSO aware if it has been disabled.

Note You may want to disable SSO HSRP by using the **no standby sso** command if you have LAN segments that should switch HSRP traffic to a redundant device while SSO maintains traffic flow for other connections.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **mode sso**
5. **exit**
6. **no standby sso**
7. **standby sso**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**mode sso** **Example:** `Device(config-red)# mode sso`	Enables the redundancy mode of operation to SSO. • HSRP is SSO aware on interfaces that are configured for HSRP and the standby RP is automatically reset.
Step 5	**exit** **Example:** `Device(config-red)# exit`	Exits redundancy configuration mode.
Step 6	**no standby sso** **Example:** `Device(config)# no standby sso`	Disables HSRP SSO mode for all HSRP groups.
Step 7	**standby sso** **Example:** `Device(config)# standby sso`	Enables the SSO HSRP feature if you have disabled the functionality.
Step 8	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.

Verifying SSO Aware HSRP

To verify or debug HSRP SSO operation, perform the following steps from the active RP console.

SUMMARY STEPS

1. **show standby**
2. **debug standby events ha**

DETAILED STEPS

Step 1 show standby

Use the **show standby** command to display the state of the standby RP, for example:

Example:

```
Device# show standby

GigabitEthernet0/0/0 - Group 1
 State is Active (standby RP)
 Virtual IP address is 10.1.0.7
 Active virtual MAC address is unknown
  Local virtual MAC address is 000a.f3fd.5001 (bia)
 Hello time 1 sec, hold time 3 sec
 Authentication text "authword"
 Preemption enabled
 Active router is unknown
 Standby router is unknown
 Priority 110 (configured 120)
  Track object 1 state Down decrement 10
 Group name is "name1" (cfgd)
```

Step 2 debug standby events ha

Use the**debug standby events ha** command to display the active and standby RPs, for example:

Example:

```
Device# debug standby events ha

!Active RP
*Apr 27 04:13:47.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Listen into sync buffer
*Apr 27 04:13:47.855: HSRP: CF Sync send ok
*Apr 27 04:13:57.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Speak into sync buffer
*Apr 27 04:13:57.855: HSRP: CF Sync send ok
*Apr 27 04:14:07.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Standby into sync buffer
*Apr 27 04:14:07.755: HSRP: Gi0/0/1 Grp 101 RF Encode state Active into sync buffer
*Apr 27 04:14:07.863: HSRP: CF Sync send ok
*Apr 27 04:14:07.867: HSRP: CF Sync send ok
!Standby RP
*Apr 27 04:11:21.011: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:21.011: HSRP: Gi0/0/1 Grp 101 RF sync state Init -> Listen
*Apr 27 04:11:31.011: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:31.011: HSRP: Gi0/0/1 Grp 101 RF sync state Listen -> Speak
*Apr 27 04:11:41.071: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:41.071: HSRP: RF CF client 32, entity 0 got msg len 24
*Apr 27 04:11:41.071: HSRP: Gi0/0/1 Grp 101 RF sync state Speak -> Standby
*Apr 27 04:11:41.071: HSRP: Gi0/0/1 Grp 101 RF sync state Standby -> Active
```

Configuration Examples for SSO HSRP

Example: Enabling SSO-Aware HSRP

The following example shows how to set the redundancy mode to SSO. HSRP is automatically SSO-aware when this mode is enabled.

```
Device(config)# redundancy
Device(config-red)# mode sso
```

If SSO HSRP is disabled using the **no standby sso** command, you can reenable it as shown in the following example:

```
Device(config)# interface GigabitEthernet 1/0/0
Device(config-if)# ip address 10.1.1.1 255.255.0.0
Device(config-if)# standby priority 200
Device(config-if)# standby preempt
Device(config-if)# standby sso
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	"HSRP for IPv6" module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

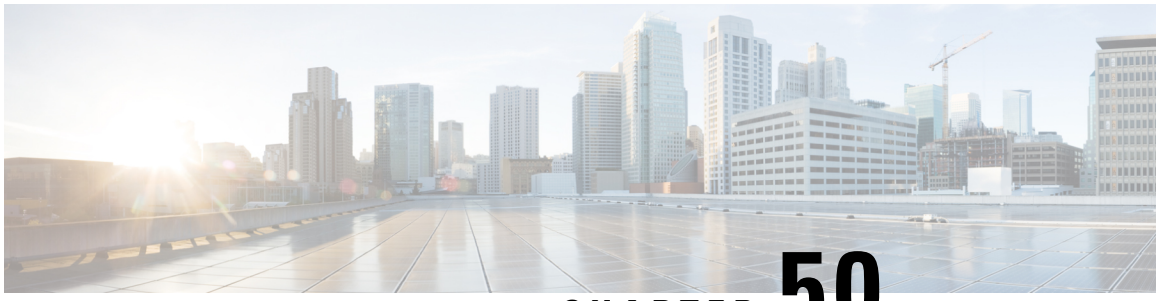
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for SSO - HSRP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 50

HSRP - ISSU

Information About HSRP - ISSU

HSRP—ISSU

The In Service Software Upgrade (ISSU) process allows Cisco software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco software to be modified while packet forwarding continues, which increases network availability and reduces downtime caused by planned software upgrades.

For detailed information about ISSU, see the *Cisco IOS In Service Software Upgrade Process* document in the *High Availability Configuration Guide*.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	"HSRP for IPv6" module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

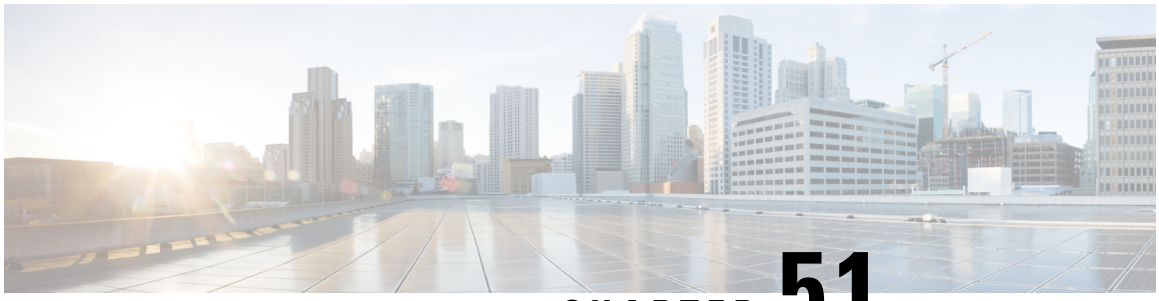
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HSRP - ISSU

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 51

FHRP - HSRP MIB

Information About FHRP - HSRP MIB

HSRP MIB Traps

HSRP MIB supports Simple Network Management Protocol (SNMP) Get operations, to allow network devices to get reports about HSRP groups in a network from the network management station.

Enabling HSRP MIB trap support is performed through the CLI, and the MIB is used for getting the reports. A trap notifies the network management station when a device leaves or enters the active or standby state. When an entry is configured from the CLI, the RowStatus for that group in the MIB immediately goes to the active state.

Cisco software supports a read-only version of the MIB, and set operations are not supported.

This functionality supports four MIB tables, as follows:

- cHsrpGrpEntry table defined in CISCO-HSRP-MIB.my
- cHsrpExtIfTrackedEntry, defined in CISCO-HSRP-EXT-MIB.my
- cHsrpExtSecAddrEntry, defined in CISCO-HSRP-EXT-MIB.my
- cHsrpExtIfEntry defined in CISCO-HSRP-EXT-MIB.my

The cHsrpGrpEntry table consists of all the group information defined in RFC 2281, *Cisco Hot Standby Router Protocol*; the other tables consist of the Cisco extensions to RFC 2281, which are defined in CISCO-HSRP-EXT-MIB.my.

How to Configure FHRP - HSRP MIB

Enabling HSRP MIB Traps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps hsrp**
4. **snmp-server host** *host community-string* **hsrp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps hsrp** **Example:** `Device(config)# snmp-server enable traps hsrp`	Enables the device to send SNMP traps and informs, and HSRP notifications.
Step 4	**snmp-server host** *host community-string* **hsrp** **Example:** `Device(config)# snmp-server host myhost.comp.com public hsrp`	Specifies the recipient of an SNMP notification operation, and that HSRP notifications be sent to the host.

Configuration Examples for FHRP - HSRP MIB

Example: Enabling HSRP MIB Traps

The following examples show how to configure HSRP on two devices and enable the HSRP MIB trap support functionality. As in many environments, one device is preferred as the active one. To configure a device's preference as the active device, configure the device at a higher priority level and enable preemption. In the

following example, the active device is referred to as the primary device. The second device is referred to as the backup device:

Device A

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.1.1.1 255.255.0.0
Device(config-if)# standby priority 200
Device(config-if)# standby preempt
Device(config-if)# standby ip 10.1.1.3
Device(config-if)# exit
Device(config)# snmp-server enable traps hsrp
Device(config)# snmp-server host yourhost.cisco.com public hsrp
```

Device B

```
Device(config)#interface GigabitEthernet 1/0/0
Device(config-if)# ip address 10.1.1.2 255.255.0.0
Device(config-if)#  standby priority 101
Device(config-if)# standby ip 10.1.1.3
Device(config-if)# exit
Device(config)# snmp-server enable traps hsrp
Device(config)# snmp-server host myhost.cisco.com public hsrp
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	“HSRP for IPv6” module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

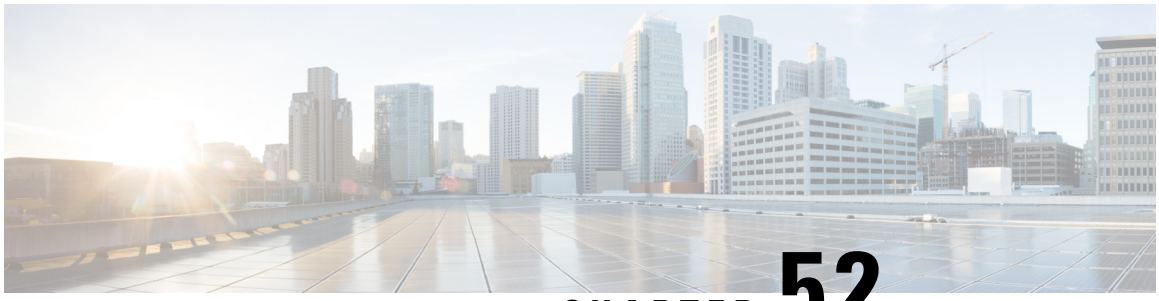
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for FHRP - HSRP-MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 52

HSRP Support for MPLS VPNs

Information About HSRP Support for MPLS VPNs

HSRP Support for MPLS VPNs

HSRP support for a Multiprotocol Label Switching (MPLS) VPN interface is useful when an Ethernet LAN is connected between two provider edge (PE) devices with either of the following conditions:

- A customer edge (CE) device with a default route to the HSRP virtual IP address
- One or more hosts with the HSRP virtual IP address configured as the default gateway

Each VPN is associated with one or more VPN routing and forwarding (VRF) instances. A VRF consists of the following elements:

- IP routing table
- Cisco Express Forwarding table
- Set of interfaces that use the Cisco Express Forwarding forwarding table
- Set of rules and routing protocol parameters to control the information in the routing tables

VPN routing information is stored in the IP routing table and the Cisco Express Forwarding table for each VRF. A separate set of routing and Cisco Express Forwarding tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN and also prevent packets that are outside a VPN from being forwarded to a device within the VPN.

HSRP adds ARP entries and IP hash table entries (aliases) using the default routing table instance. However, a different routing table instance is used when VRF forwarding is configured on an interface, causing ARP and ICMP echo requests for the HSRP virtual IP address to fail.

HSRP support for MPLS VPNs ensures that the HSRP virtual IP address is added to the correct IP routing table and not to the default routing table.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop redundancy Protocols Command Reference*
HSRP for IPv6	"HSRP for IPv6" module
Troubleshooting HSRP	Hot Standby Router Protocol: Frequently Asked Questions

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
CISCO-HSRP-MIB CISCO-HSRP-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 2281	*Cisco Hot Standby Router Protocol*

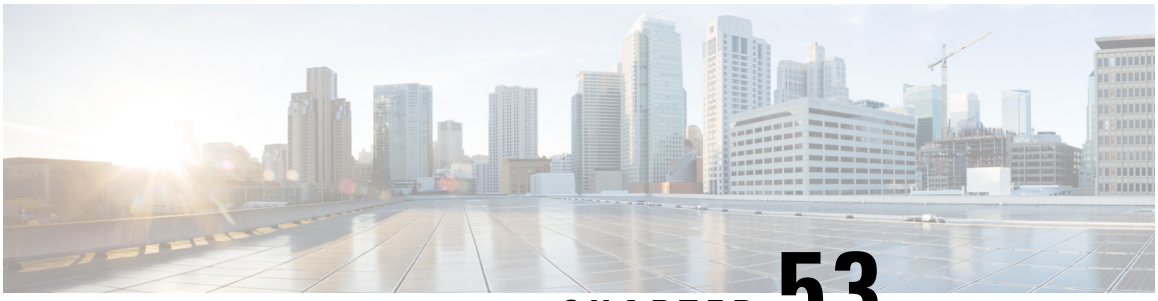
Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for HSRP Support for MPLS VPNs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.



CHAPTER 53

Configuring VRRP

The Virtual Router Redundancy Protocol (VRRP) is an election protocol that dynamically assigns responsibility for one or more virtual routers to the VRRP routers on a LAN, allowing several routers on a multiaccess link to utilize the same virtual IP address. A VRRP router is configured to run the VRRP protocol in conjunction with one or more other routers attached to a LAN. In a VRRP configuration, one router is elected as the virtual primary router, with the other routers acting as backups in case the virtual primary router fails.

This module explains the concepts related to VRRP and describes how to configure VRRP in a network.

Restrictions for VRRP

- VRRP is designed for use over multiaccess, multicast, or broadcast capable Ethernet LANs. VRRP is not intended as a replacement for existing dynamic protocols.
- VRRP is supported on Ethernet, Fast Ethernet, Bridge Group Virtual Interface (BVI), and Gigabit Ethernet interfaces, and on Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs), VRF-aware MPLS VPNs, and VLANs.
- Because of the forwarding delay that is associated with the initialization of a BVI interface, you must configure the VRRP advertise timer to a value equal to or greater than the forwarding delay on the BVI interface. This setting prevents a VRRP router on a recently initialized BVI interface from unconditionally taking over the primary role. Use the **bridge forward-time** command to set the forwarding delay on the BVI interface. Use the **vrrp timers advertise** command to set the VRRP advertisement timer.

Information About VRRP

VRRP Operation

There are several ways a LAN client can determine which router should be the first hop to a particular remote destination. The client can use a dynamic process or static configuration. Examples of dynamic router discovery are as follows:

- Proxy ARP—The client uses Address Resolution Protocol (ARP) to get the destination it wants to reach, and a router will respond to the ARP request with its own MAC address.
- Routing protocol—The client listens to dynamic routing protocol updates (for example, from Routing Information Protocol [RIP]) and forms its own routing table.
- ICMP Router Discovery Protocol (IRDP) client—The client runs an Internet Control Message Protocol (ICMP) router discovery client.

The drawback to dynamic discovery protocols is that they incur some configuration and processing overhead on the LAN client. Also, in the event of a router failure, the process of switching to another router can be slow.

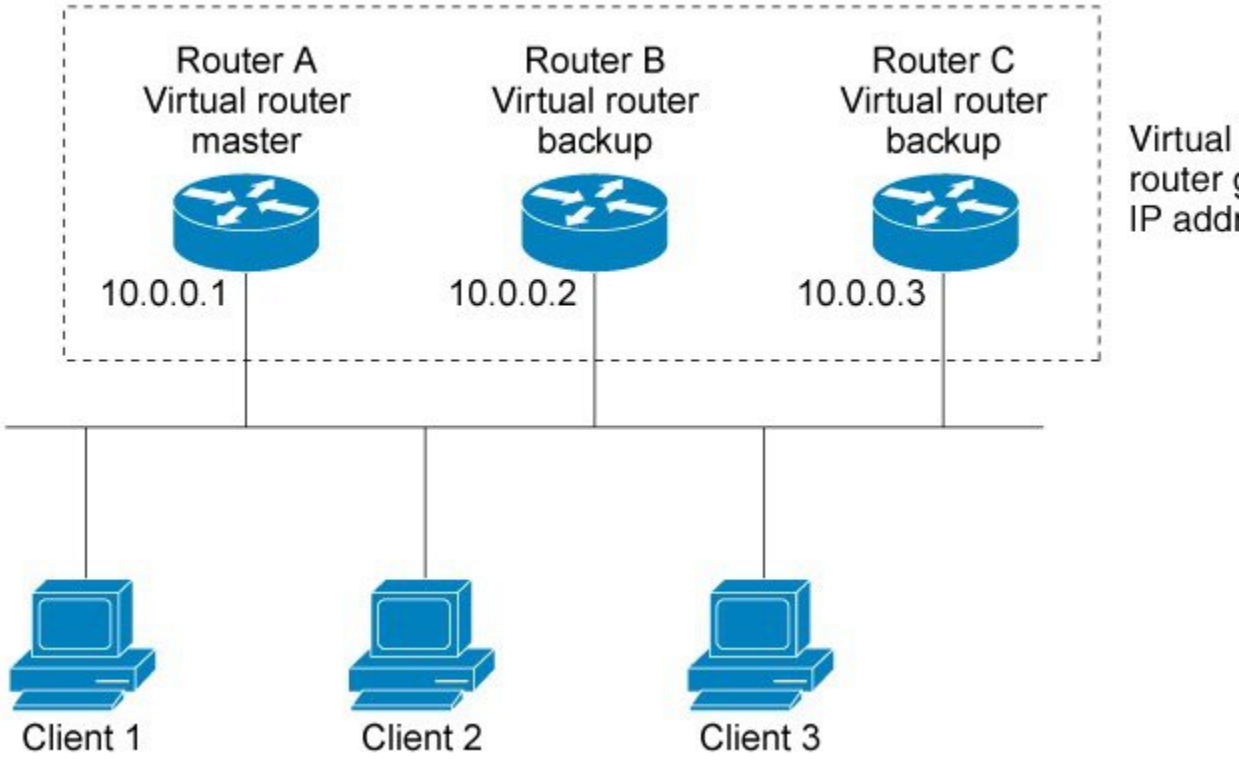
An alternative to dynamic discovery protocols is to statically configure a default router on the client. This approach simplifies client configuration and processing, but creates a single point of failure. If the default gateway fails, the LAN client is limited to communicating only on the local IP network segment and is cut off from the rest of the network.

VRRP can solve the static configuration problem. VRRP enables a group of routers to form a single *virtual router*. The LAN clients can then be configured with the virtual router as their default gateway. The virtual router, representing a group of routers, is also known as a VRRP group.

VRRP is supported on Ethernet, Fast Ethernet, BVI, and Gigabit Ethernet interfaces, and on MPLS VPNs, VRF-aware MPLS VPNs, and VLANs.

The figure below shows a LAN topology in which VRRP is configured. In this example, Routers A, B, and C are VRRP routers (routers running VRRP) that comprise a virtual router. The IP address of the virtual router is the same as that configured for the Ethernet interface of Router A (10.0.0.1).

Figure 16: Basic VRRP Topology

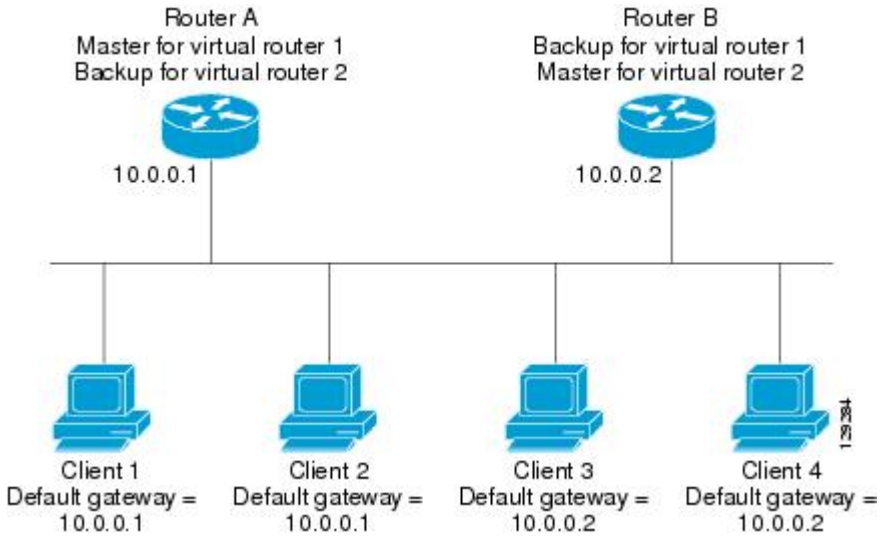


Because the virtual router uses the IP address of the physical Ethernet interface of Router A, Router A assumes the role of the virtual primary router and is also known as the IP address owner. As the virtual primary router, Router A controls the IP address of the virtual router and is responsible for forwarding packets sent to this IP address. Clients 1 through 3 are configured with the default gateway IP address of 10.0.0.1.

Routers B and C function as virtual router backups. If the virtual primary router fails, the router configured with the higher priority will become the virtual primary router and provide uninterrupted service for the LAN hosts. When Router A recovers, it becomes the virtual primary router again. For more detail on the roles that VRRP routers play and what happens if the virtual primary router fails, see the VRRP Router Priority and Preemption section.

The figure below shows a LAN topology in which VRRP is configured so that Routers A and B share the traffic to and from clients 1 through 4 and that Routers A and B act as virtual router backups to each other if either router fails.

Figure 17: Load Sharing and Redundancy VRRP Topology



In this topology, two virtual routers are configured. (For more information, see the Multiple Virtual Router Support section.) For virtual router 1, Router A is the owner of IP address 10.0.0.1 and virtual primary router, and Router B is the virtual router backup to Router A. Clients 1 and 2 are configured with the default gateway IP address of 10.0.0.1.

For virtual router 2, Router B is the owner of IP address 10.0.0.2 and virtual primary router, and Router A is the virtual router backup to Router B. Clients 3 and 4 are configured with the default gateway IP address of 10.0.0.2.

VRRP Benefits

Redundancy

VRRP enables you to configure multiple routers as the default gateway router, which reduces the possibility of a single point of failure in a network.

Load Sharing

You can configure VRRP in such a way that traffic to and from LAN clients can be shared by multiple routers, thereby sharing the traffic load more equitably among available routers.

Multiple Virtual Routers

Multiple IP Addresses

The virtual router can manage multiple IP addresses, including secondary IP addresses. Therefore, if you have multiple subnets configured on an Ethernet interface, you can configure VRRP on each subnet.

Preemption

The redundancy scheme of VRRP enables you to preempt a virtual router backup that has taken over for a failing virtual primary router with a higher priority virtual router backup that has become available.

Authentication

VRRP message digest 5 (MD5) algorithm authentication protects against VRRP-spoofing software and uses the industry-standard MD5 algorithm for improved reliability and security.

Advertisement Protocol

VRRP uses a dedicated Internet Assigned Numbers Authority (IANA) standard multicast address (224.0.0.18) for VRRP advertisements. This addressing scheme minimizes the number of routers that must service the multicasts and allows test equipment to accurately identify VRRP packets on a segment. The IANA assigned VRRP the IP protocol number 112.

VRRP Object Tracking

VRRP object tracking provides a way to ensure the best VRRP router is the virtual primary router for the group by altering VRRP priorities to the status of tracked objects such as the interface or IP route states.

Multiple Virtual Router Support

- Router processing capability
- Router memory capability
- Router interface support of multiple MAC addresses

In a topology where multiple virtual routers are configured on a router interface, the interface can act as primary for one virtual router and as a backup for one or more virtual routers.

VRRP Router Priority and Preemption

An important aspect of the VRRP redundancy scheme is VRRP router priority. Priority determines the role that each VRRP router plays and what happens if the virtual primary router fails.

If a VRRP router owns the IP address of the virtual router and the IP address of the physical interface, this router will function as a virtual primary router.

Priority also determines if a VRRP router functions as a virtual router backup and the order of ascendancy to becoming virtual primary router if the virtual primary router fails. You can configure the priority of each virtual router backup with a value of 1 through 254 using the **vrrp priority** command.

For example, if Router A, the virtual primary router in a LAN topology, fails, an election process takes place to determine if virtual router backups B or C should take over. If Routers B and C are configured with the priorities of 101 and 100, respectively, Router B is elected to become virtual primary router because it has the higher priority. If Routers B and C are both configured with the priority of 100, the virtual router backup with the higher IP address is elected to become the virtual primary router.

By default, a preemptive scheme is enabled whereby a higher priority virtual router backup that becomes available takes over for the virtual router backup that was elected to become virtual primary router. You can disable this preemptive scheme using the **no vrrp preempt** command. If preemption is disabled, the virtual router backup that is elected to become virtual primary router remains as the primary until the original virtual primary router recovers and becomes the primary again.

VRRP Advertisements

The virtual primary router sends VRRP advertisements to other VRRP routers in the same group. The advertisements communicate the priority and state of the virtual primary router. The VRRP advertisements are encapsulated in IP packets and sent to the IP Version 4 multicast address assigned to the VRRP group. The advertisements are sent every second by default; the interval is configurable.

Although the VRRP protocol as per RFC 3768 does not support millisecond timers, Cisco routers allow you to configure millisecond timers. You need to manually configure the millisecond timer values on both the primary and the backup routers. The primary advertisement value displayed in the **show vrrp** command output on the backup routers is always 1 second because the packets on the backup routers do not accept millisecond values.

You must use millisecond timers where absolutely necessary and with careful consideration and testing. Millisecond values work only under favorable circumstances, and you must be aware that the use of the millisecond timer values restricts VRRP operation to Cisco devices only.

VRRP Object Tracking

Object tracking is an independent process that manages creating, monitoring, and removing tracked objects such as the state of the line protocol of an interface. Clients such as the Hot Standby Router Protocol (HSRP), Gateway Load Balancing Protocol (GLBP), and VRRP register their interest with specific tracked objects and act when the state of an object changes.

Each tracked object is identified by a unique number that is specified on the tracking CLI. Client processes such as VRRP use this number to track a specific object.

The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

VRRP object tracking gives VRRP access to all the objects available through the tracking process. The tracking process allows you to track individual objects such as a the state of an interface line protocol, state of an IP route, or the reachability of a route.

VRRP provides an interface to the tracking process. Each VRRP group can track multiple objects that may affect the priority of the VRRP device. You specify the object number to be tracked and VRRP is notified of any change to the object. VRRP increments (or decrements) the priority of the virtual device based on the state of the object being tracked.

How VRRP Object Tracking Affects the Priority of a Device

The priority of a device can change dynamically if it has been configured for object tracking and the object that is being tracked goes down. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to VRRP, either immediately or after a specified delay. The object values are reported as either up or down. Examples of objects that can be tracked are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, the VRRP priority is reduced. The VRRP device with the higher priority can now become the virtual primary device if it has the **vrrp preempt** command configured. See the “VRRP Object Tracking” section for more information on object tracking.

In Service Software Upgrade--VRRP

VRRP supports In Service Software Upgrade (ISSU). In Service Software Upgrade (ISSU) allows a high-availability (HA) system to run in stateful switchover (SSO) mode even when different versions of Cisco IOS XE software are running on the active and standby Route Processors (RPs) or line cards.

ISSU provides the ability to upgrade or downgrade from one supported Cisco IOS XE release to another while continuing to forward packets and maintain sessions, thereby reducing planned outage time. The ability to upgrade or downgrade is achieved by running different software versions on the active RP and standby RP for a short period of time to maintain state information between RPs. This feature allows the system to switch over to a secondary RP running upgraded (or downgraded) software and continue forwarding packets without session loss and with minimal or no packet loss. This feature is enabled by default.

For detailed information about ISSU, see the Cisco IOS XE In Service Software Upgrade Process document in the *Cisco IOS XE High Availability Configuration Guide*.

VRRP Support for Stateful Switchover

With the introduction of the VRRP Support for Stateful Switchover feature, VRRP is SSO aware. VRRP can detect when a router is failing over to the secondary RP and continue in its current group state.

SSO functions in networking devices (usually edge devices) that support dual Route Processors (RPs). SSO provides RP redundancy by establishing one of the RPs as the active processor and the other RP as the standby processor. SSO also synchronizes critical state information between the RPs so that network state information is dynamically maintained between RPs.

Prior to being SSO aware, if VRRP was deployed on a router with redundant RPs, a switchover of roles between the active RP and the standby RP would result in the router relinquishing its activity as a VRRP group member and then rejoining the group as if it had been reloaded. The SSO--VRRP feature enables VRRP to continue its activities as a group member during a switchover. VRRP state information between redundant RPs is maintained so that the standby RP can continue the router's activities within the VRRP during and after a switchover.

This feature is enabled by default. To disable this feature, use the **no vrrp sso** command in global configuration mode.

For more information, see the Stateful Switchover document.

How to Configure VRRP

VRRP

Customizing the behavior of VRRP is optional. Be aware that as soon as you enable a VRRP group, that group is operating. It is possible that if you first enable a VRRP group before customizing VRRP, the router could take over control of the group and become the virtual primary router before you have finished customizing the feature. Therefore, if you plan to customize VRRP, it is a good idea to do so before enabling VRRP.

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Router(config)#GigabitEthernet 0/0/0
```

Enters interface configuration mode.

Step 4 **ip address** *ip-address mask*

Example:

```
Router(config-if)# ip address 172.16.6.5 255.255.255.0
```

Configures an IP address for an interface.

Step 5 **vrrp** *group* **description** *text*

Example:

```
Router(config-if)# vrrp 10 description working-group
```

Assigns a text description to the VRRP group.

Step 6 **vrrp** *group* **priority** *level*

Example:

```
Router(config-if)# vrrp 10 priority 110
```

Sets the priority level of the router within a VRRP group.

- The default priority is 100.

Step 7 **vrrp** *group* **preempt** [**delay minimum** *seconds*]

Example:

```
Router(config-if)# vrrp 10 preempt delay minimum 380
```

Configures the router to take over as virtual primary router for a VRRP group if it has a higher priority than the current virtual primary router.

- The default delay period is 0 seconds.
- The router that is IP address owner will preempt, regardless of the setting of this command.

Step 8 **vrrp** *group* **timers learn**

Example:

```
Router(config-if)# vrrp 10 timers learn
```

Configures the router, when it is acting as virtual router backup for a VRRP group, to learn the advertisement interval used by the virtual primary router.

Step 9 **exit**

Example:

```
Router(config-if)# exit
```

Exits interface configuration mode.

Step 10 **no vrrp sso**

Example:

```
Router(config)# no vrrp sso
```

(Optional) Disables VRRP support of SSO.

- VRRP support of SSO is enabled by default.

EnablingVerifying VRRP

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Router(config)# interfaceGigabitEthernet 0/0/0
```

Enters interface configuration mode.

Step 4 **ip address** *ip-address mask*

Example:

```
Router(config-if)# ip address 172.16.6.5 255.255.255.0
```

Configures an IP address for an interface.

Step 5 **vrrp** *group* **ip** *ip-address* [**secondary**]

Example:

```
Router(config-if)# vrrp 10 ip 172.16.6.1
```

Enables VRRP on an interface.

- After you identify a primary IP address, you can use the **vrrp ip** command again with the **secondary** keyword to indicate additional IP addresses supported by this group.

Note All routers in the VRRP group must be configured with the same primary address and a matching list of secondary addresses for the virtual router. If different primary or secondary addresses are configured, the routers in the VRRP group will not communicate with each other and any misconfigured router will change its state to primary.

Step 6 **show vrrp** [**brief all**] | interface]

Example:

```
Router(config-if)#show vrrp brief
Interface Grp Pri Time Own Pre State Master addr Group addr
BD10 1 100 9609 Y Backup 10.1.0.2 10.1.0.10
BD10 5 200 90218 Y Master 10.1.0.1 10.1.0.50
BD10 100 100 3609 Backup 10.1.0.2 10.1.0.100
```

(Optional) Displays a brief or detailed status of one or all VRRP groups on the router.

Step 7 **show vrrp interface** *type number* [**brief**]

Example:

```
Router(config)# interfaceGigabitEthernet 0/0/0
Router)config-if)#show vrrp interface bdi10
BDI10 - Group 10
G1
State is Master
Virtual IP address is 10.0.0.5
Virtual MAC address is 0000.5e00.010a
Advertisement interval is 10.000 sec
Preemption enabled, delay min 380 secs
Priority is 110
Master Router is 10.0.0.2 (local), priority is 110
Master Advertisement interval is 10.000 sec
Master Down interval is 30.570 sec
FLAGS: 1/1
```

(Optional) Displays the VRRP groups and their status on a specified interface.

Step 8 **end**

Example:

```
Router(config-if)# end
```

Returns to privileged EXEC mode.

Configuring VRRP Object Tracking

Note If a VRRP group is the IP address owner, its priority is fixed at 255 and cannot be reduced through object tracking.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **interface** *type number* {**line-protocol** | **ip routing**}
4. **interface** *type number*
5. **vrrp** *group* **ip** *ip-address*
6. **vrrp** *group* **priority** *level*
7. **vrrp** *group* **track** *object-number* [**decrement** *priority*]
8. **end**
9. **show track** [*object-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**track** *object-number* **interface** *type number* {**line-protocol** | **ip routing**} **Example:** `Router(config)# track 2 interface serial 6 line-protocol`	Configures an interface to be tracked where changes in the state of the interface affect the priority of a VRRP group. • This command configures the interface and corresponding object number to be used with the **vrrp track** command. • The **line-protocol** keyword tracks whether the interface is up. The **ip routing** keyword also checks that IP routing is enabled and active on the interface. • You can also use the **track ip route** command to track the reachability of an IP route or a metric type object.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet 2`	Enters interface configuration mode.
Step 5	**vrrp** *group* **ip** *ip-address* **Example:** `Router(config-if)# vrrp 1 ip 10.0.1.20`	Enables VRRP on an interface and identifies the IP address of the virtual router.
Step 6	**vrrp** *group* **priority** *level* **Example:** `Router(config-if)# vrrp 1 priority 120`	Sets the priority level of the router within a VRRP group.
Step 7	**vrrp** *group* **track** *object-number* [**decrement** *priority*] **Example:** `Router(config-if)# vrrp 1 track 2 decrement 15`	Configures VRRP to track an object.
Step 8	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 9	**show track** [*object-number*] **Example:** `Router# show track 1`	Displays tracking information.

Configuring VRRP Text Authentication

Before you begin

Interoperability with vendors that may have implemented the RFC 2338 method is not enabled.

Text authentication cannot be combined with MD5 authentication for a VRRP group at any one time. When MD5 authentication is configured, the text authentication field in VRRP hello messages is set to all zeros on transmit and ignored on receipt, provided the receiving router also has MD5 authentication enabled.

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **terminal interface** *type number*

Example:

```
Router(config)# interfaceGigabitEthernet 0/0/0 Ethernet 0/1
```

Configures an interface type and enters interface configuration mode.

Step 4 **ip address** *ip-address mask* [**secondary**]

Example:

```
Router(config-if)# ip address 10.0.0.1 255.255.255.0
```

Specifies a primary or secondary IP address for an interface.

Step 5 **vrrp** *group* **authentication text** *text-string*

Example:

```
Router(config-if)# vrrp 1 authentication text textstring1
```

Authenticates VRRP packets received from other routers in the group.

- If you configure authentication, all routers within the VRRP group must use the same authentication string.
- The default string is cisco.

Note All routers within the VRRP group must be configured with the same authentication string. If the same authentication string is not configured, the routers in the VRRP group will not communicate with each other and any misconfigured router will change its state to primary.

Step 6 **vrrp** *group* **ip** *ip-address*

Example:

```
Router(config-if)# vrrp 1 ip 10.0.1.20
```

Enables VRRP on an interface and identifies the IP address of the virtual router.

Step 7 Repeat Steps 1 through 6 on each router that will communicate.

—

Step 8 **end**

Example:

```
Router(config-if)# end
```

Returns to privileged EXEC mode.

Configuration Examples for VRRPv2

Example: Configuring VRRP

In the following example, Router A and Router B each belong to three VRRP groups.

In the configuration, each group has the following properties:

- Group 1:
  - Virtual IP address is 10.1.0.10.
  - Router A will become the primary for this group with priority 120.
  - Advertising interval is 3 seconds.
  - Preemption is enabled.
- Group 5:
  - Router B will become the primary for this group with priority 200.
  - Advertising interval is 30 seconds.
  - Preemption is enabled.
- Group 100:
  - Router A will become the primary for this group first because it has a higher IP address (10.1.0.2).
  - Advertising interval is the default 1 second.
  - Preemption is disabled.

Router A

```
Router(config)#
Router(config)# interfaceGigabitEthernet 0/0/0interface GigabitEthernet 1/0/0
Router(config-if)# ip address 10.1.0.2 255.0.0.0
Router(config-if)# vrrp 1 priority 120
Router(config-if)# vrrp 1 authentication text cisco
Router(config-if)# vrrp 1 timers advertise 3
Router(config-if)# vrrp 1 timers learn
Router(config-if)# vrrp 1 ip 10.1.0.10
Router(config-if)# vrrp 5 priority 100
Router(config-if)# vrrp 5 timers advertise 30
Router(config-if)# vrrp 5 timers learn
Router(config-if)# vrrp 5 ip 10.1.0.50
Router(config-if)# vrrp 100 timers learn
Router(config-if)# no vrrp 100 preempt
Router(config-if)# vrrp 100 ip 10.1.0.100
Router(config-if)# no shutdown
```

Router B

```
Router(config)#GigabitEthernet 0/0/0interface GigabitEthernet 1/0/0
Router(config-if)# ip address 10.1.0.1 255.0.0.0
Router(config-if)# vrrp 1 priority 100
Router(config-if)# vrrp 1 authentication text cisco
Router(config-if)# vrrp 1 timers advertise 3
Router(config-if)# vrrp 1 timers learn
Router(config-if)# vrrp 1 ip 10.1.0.10
Router(config-if)# vrrp 5 priority 200
Router(config-if)# vrrp 5 timers advertise 30
Router(config-if)# vrrp 5 timers learn
Router(config-if)# vrrp 5 ip 10.1.0.50
Router(config-if)# vrrp 100 timers learn
Router(config-if)# no vrrp 100 preempt
Router(config-if)# vrrp 100 ip 10.1.0.100
Router(config-if)# no shutdown
```

Example: VRRP Object Tracking

In the following example, the tracking process is configured to track the state of the line protocol on serial interface 0/1. VRRP on Ethernet interface 1/0 then registers with the tracking process to be informed of any changes to the line protocol state of serial interface 0/1. If the line protocol state on serial interface 0/1 goes down, then the priority of the VRRP group is reduced by 15.

```
Router(config)# track 1 interface Serial 0/1 line-protocol
Router(config-track)# exit
Router(config)# interface Ethernet 1/0
Router(config-if)# ip address 10.0.0.2 255.0.0.0
Router(config-if)# vrrp 1 ip 10.0.0.3
Router(config-if)# vrrp 1 priority 120
Router(config-if)# vrrp 1 track 1 decrement 15
```

Example: VRRP Object Tracking Verification

The following examples verify the configuration shown in the Example: VRRP Object Tracking section:

```
Router# show vrrp

Ethernet1/0 - Group 1
  State is Master
  Virtual IP address is 10.0.0.3
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000 sec
  Preemption is enabled
   min delay is 0.000 sec
  Priority is 105
   Track object 1 state Down decrement 15
  Master Router is 10.0.0.2 (local), priority is 105
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.531 sec
Router# show track

Track 1
  Interface Serial0/1 line-protocol
  Line protocol is Down (hw down)
   1 change, last change 00:06:53
```

```
Tracked by:
 VRRP Ethernet1/0 1
```

Example: VRRP Text Authentication

The following example shows how to configure VRRP text authentication using a text string:

```
Router(config)#GigabitEthernet 0/0/0interface GigabitEthernet 0/0/0
Router(config)# ip address 10.21.8.32 255.255.255.0
Router(config-if)# vrrp 10 authentication text stringxyz
Router(config-if)# vrrp 10 ip 10.21.8.10
```

Example: VRRP MIB Trap

```
Router(config)# snmp-server enable traps vrrp
Router(config)# snmp-server host 10.1.1.0 community abc vrrp
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
VRRP commands	*Cisco IOS IP Application Services Command Reference*
Object tracking	Configuring Enhanced Object Tracking
Hot Standby Routing Protocol (HSRP)	Configuring HSRP
In Service Software Upgrace (ISSU)	"In Service Software Upgrade Process" in the *High Availability Configuration Guide*
Gateway Load Balancing Protocol (GLBP)	Configuring GLBP
Stateful Switchover	The Stateful Switchover section in the*High Availability Configuration Guide*

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
VRRP MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2338	Virtual Router Redundancy Protocol
RFC 2787	Definitions of Managed Objects for the Virtual Router Redundancy Protocol
RFC 3768	Virtual Router Redundancy Protocol (VRRP)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VRRP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Glossary

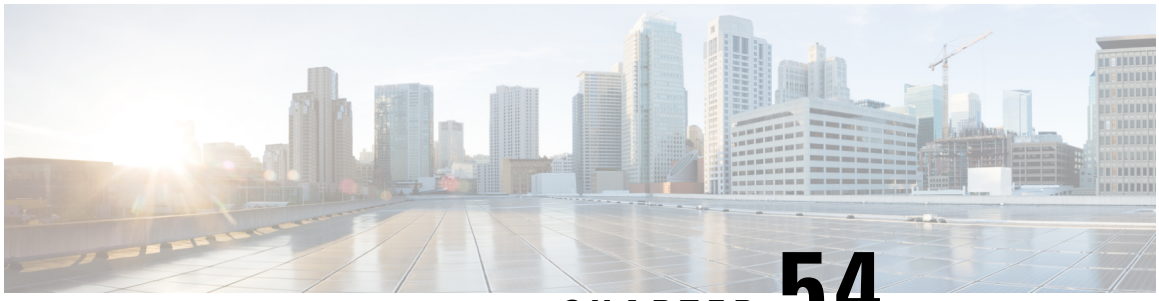
virtual IP address owner —The VRRP router that owns the IP address of the virtual router. The owner is the router that has the virtual router address as its physical interface address.

virtual router —One or more VRRP routers that form a group. The virtual router acts as the default gateway router for LAN clients. Also known as a VRRP group.

virtual router backup —One or more VRRP routers that are available to assume the role of forwarding packets if the virtual primary router fails.

virtual primary router —The VRRP router that is currently responsible for forwarding packets sent to the IP addresses of the virtual router. Usually the virtual primary router also functions as the IP address owner.

VRRP router --A router that is running VRRP.



CHAPTER 54

VRRPv3 Protocol Support

Virtual Router Redundancy Protocol (VRRP) enables a group of devices to form a single virtual device to provide redundancy. The LAN clients can then be configured with the virtual device as their default gateway. The virtual device, representing a group of devices, is also known as a VRRP group. The VRRP version 3 (v3) Protocol Support feature provides the capability to support IPv4 and IPv6 addresses while VRRP version 2 (v2) only supports IPv4 addresses. This module explains concepts related to VRRPv3 and describes how to create and customize a VRRP group in a network. Benefits of using VRRPv3 Protocol Support include the following:

- Interoperability in multi-vendor environments.
- VRRPv3 supports usage of IPv4 and IPv6 addresses while VRRPv2 only supports IPv4 addresses.
- Improved scalability through the use of VRRS Pathways.

Note In this module, VRRP and VRRPv3 are used interchangeably.

Restrictions for VRRPv3 Protocol Support

- VRRPv3 is not intended as a replacement for existing dynamic protocols. VRRPv3 is designed for use over multi-access, multicast, or broadcast capable Ethernet LANs.
- VRRPv3 is supported on Ethernet, Fast Ethernet, Bridge Group Virtual Interface (BVI), and Gigabit Ethernet interfaces, and on Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs), VRF-aware MPLS VPNs, and VLANs.
- Because of the forwarding delay that is associated with the initialization of a BVI interface, you must not configure the VRRPv3 advertise timer to a value lesser than the forwarding delay on the BVI interface.

If you configure the VRRPv3 advertise timer to a value equal to or greater than the forwarding delay on the BVI interface, the setting prevents a VRRP device on a recently initialized BVI interface from unconditionally taking over the primary role. Use the **bridge forward-time** command to set the forwarding delay on the BVI interface. Use the **vrrp timers advertise** command to set the VRRP advertisement timer.

- VRRPv3 does not support Stateful Switchover (SSO).
- VRRPv3 protocol does not support authentication.
- Full network redundancy can only be achieved if VRRP operates over the same network path as the VRRS Pathway redundant interfaces. For full redundancy, the following restrictions apply:
  - VRRS pathways should not share a different physical interface as the parent VRRP group or be configured on a sub-interface having a different physical interface as the parent VRRP group.
  - VRRS pathways should not be configured on Switch Virtual Interface (SVI) interfaces as long as the associated VLAN does not share the same trunk as the VLAN on which the parent VRRP group is configured.

Information About VRRPv3 Protocol Support

VRRPv3 Benefits

Support for IPv4 and IPv6

VRRPv3 supports IPv4 and IPv6 address families while VRRPv2 only supports IPv4 addresses.

Note When VRRPv3 is in use, VRRPv2 is unavailable. For VRRPv3 to be configurable, the **fhrp version vrrp v3** command must be used in global configuration mode

Redundancy

VRRP enables you to configure multiple devices as the default gateway device, which reduces the possibility of a single point of failure in a network.

Load Sharing

You can configure VRRP in such a way that traffic to and from LAN clients can be shared by multiple devices, thereby sharing the traffic load more equitably between available devices.

Multiple Virtual Devices

VRRP supports up to 255 virtual devices (VRRP groups) on a device physical interface, subject to restrictions in scaling. Multiple virtual device support enables you to implement redundancy and load sharing in your LAN topology. In scaled environments, VRRS Pathways should be used in combination with VRRP control groups.

Multiple IP Addresses

The virtual device can manage multiple IP addresses, including secondary IP addresses. Therefore, if you have multiple subnets configured on an Ethernet interface, you can configure VRRP on each subnet.

Note To utilize secondary IP addresses in a VRRP group, a primary address must be configured on the same group.

Preemption

The redundancy scheme of VRRP enables you to preempt a virtual device backup that has taken over for a failing virtual primary device with a higher priority virtual device backup that has become available.

Note Preemption of a lower priority primary device is enabled with an optional delay.

Advertisement Protocol

VRRP uses a dedicated Internet Assigned Numbers Authority (IANA) standard multicast address for VRRP advertisements. For IPv4, the multicast address is 224.0.0.18. For IPv6, the multicast address is FF02:0:0:0:0:0:0:12. This addressing scheme minimizes the number of devices that must service the multicasts and allows test equipment to accurately identify VRRP packets on a segment. The IANA has assigned VRRP the IP protocol number 112.

VRRP Device Priority and Preemption

An important aspect of the VRRP redundancy scheme is VRRP device priority. Priority determines the role that each VRRP device plays and what happens if the virtual primary device fails.

If a VRRP device owns the IP address of the virtual device and the IP address of the physical interface, this device will function as a virtual primary device.

Priority also determines if a VRRP device functions as a virtual device backup and the order of ascendancy to becoming a virtual primary device if the virtual primary device fails. You can configure the priority of each virtual device backup with a value of 1 through 254 using the **priority** command (use the **vrrp address-family** command to enter the VRRP configuration mode and access the **priority** option).

For example, if device A, the virtual primary device in a LAN topology, fails, an election process takes place to determine if virtual device backups B or C should take over. If devices B and C are configured with the priorities of 101 and 100, respectively, device B is elected to become virtual primary device because it has the higher priority. If devices B and C are both configured with the priority of 100, the virtual device backup with the higher IP address is elected to become the virtual primary device.

By default, a preemptive scheme is enabled whereby a higher priority virtual device backup that becomes available takes over from the virtual device backup that was elected to become virtual primary device. You can disable this preemptive scheme using the **no preempt** command (use the **vrrp address-family** command to enter the VRRP configuration mode, and enter the **no preempt** command). If preemption is disabled, the virtual device backup that is elected to become virtual primary device remains the primary until the original virtual primary device recovers and becomes primary again.

Note Preemption of a lower priority primary device is enabled with an optional delay.

VRRP Advertisements

The virtual router master sends VRRP advertisements to other VRRP routers in the same group. The advertisements communicate the priority and state of the virtual router master. The VRRP advertisements are encapsulated into either IPv4 or IPv6 packets (based on the VRRP group configuration) and sent to the appropriate multicast address assigned to the VRRP group. For IPv4, the multicast address is 224.0.0.18. For IPv6, the multicast address is FF02:0:0:0:0:0:0:12. The advertisements are sent every second by default and the interval is configurable.

Cisco routers allow you to configure millisecond timers, which is a change from VRRPv2. You need to manually configure the millisecond timer values on both the primary and the backup routers. The master advertisement value displayed in the **show vrrp** command output on the backup routers is always 1 second because the packets on the backup routers do not accept millisecond values.

You must use millisecond timers where absolutely necessary and with careful consideration and testing. Millisecond values work only under favorable circumstances. The use of the millisecond timer values is compatible with third party vendors, as long as they also support VRRPv3. You can specify a timer value between 100 milliseconds and 40000 milliseconds.

How to Configure VRRPv3 Protocol Support

IPv6 VRRP Link Local Address

VRRPv3 for IPv6 requires that a primary virtual link-local IPv6 address is configured to allow the group to operate. After the primary link-local IPv6 address is established on the group, you can add the secondary global addresses.

Enabling VRRPv3 on a Device

To enable VRRPv3 on a device, perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **fhrp version vrrp v3**
4. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **fhrp version vrrp v3**

Example:

```
Device(config)# fhrp version vrrp v3
```

Enables the ability to configure VRRPv3 and VRRS.

Note When VRRPv3 is in use, VRRPv2 is unavailable.

Step 4 **end**

Example:

```
Device(config)# end
```

Returns to privileged EXEC mode.

Creating and Customizing a VRRP Group

To create a VRRP group, perform the following task. Steps 6 to 14 denote customizing options for the group, and they are optional:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **fhrp version vrrp v3**
4. **interface** *type number*
5. **vrrp** *group-id* **address-family** {**ipv4** | **ipv6**}
6. **address** *ip-address* [**primary** | **secondary**]
7. **description** *group-description*
8. **match-address**
9. **preempt delay minimum** *seconds*
10. **priority** *priority-level*
11. **timers advertise** *interval*
12. **vrrpv2**

13. **vrrs leader** *vrrs-leader-name*
14. **shutdown**
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**fhrp version vrrp v3** **Example:** `Device(config)# fhrp version vrrp v3`	Enables the ability to configure VRRPv3 and VRRS. **Note** When VRRPv3 is in use, VRRPv2 is unavailable.
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Enters interface configuration mode.
Step 5	**vrrp** *group-id* **address-family** {**ipv4** | **ipv6**} **Example:** `Device(config-if)# vrrp 3 address-family ipv4`	Creates a VRRP group and enters VRRP configuration mode.
Step 6	**address** *ip-address* [**primary** | **secondary**] **Example:** `Device(config-if-vrrp)# address 100.0.1.10 primary`	Specifies a primary or secondary address for the VRRP group. **Note** VRRPv3 for IPv6 requires that a primary virtual link-local IPv6 address is configured to allow the group to operate. After the primary link-local IPv6 address is established on the group, you can add the secondary global addresses.
Step 7	**description** *group-description* **Example:** `Device(config-if-vrrp)# description group 3`	(Optional) Specifies a description for the VRRP group.
Step 8	**match-address** **Example:**	(Optional) Matches secondary address in the advertisement packet against the configured address. • Secondary address matching is enabled by default.

	Command or Action	Purpose
	`Device(config-if-vrrp)# match-address`	
Step 9	**preempt delay minimum** *seconds* **Example:** `Device(config-if-vrrp)# preempt delay minimum 30`	(Optional) Enables preemption of lower priority primary device with an optional delay. • Preemption is enabled by default.
Step 10	**priority** *priority-level* **Example:** `Device(config-if-vrrp)# priority 3`	(Optional) Specifies the priority value of the VRRP group. • The priority of a VRRP group is 100 by default.
Step 11	**timers advertise** *interval* **Example:** `Device(config-if-vrrp)# timers advertise 1000`	(Optional) Sets the advertisement timer in milliseconds. • The advertisement timer is set to 1000 milliseconds by default.
Step 12	**vrrpv2** **Example:** `Device(config-if-vrrp)# vrrpv2`	(Optional) Enables support for VRRPv2 simultaneously, so as to interoperate with devices which only support VRRP v2. • VRRPv2 is disabled by default.
Step 13	**vrrs leader** *vrrs-leader-name* **Example:** `Device(config-if-vrrp)# vrrs leader leader-1`	(Optional) Specifies a leader's name to be registered with VRRS and to be used by followers. • A registered VRRS name is unavailable by default.
Step 14	**shutdown** **Example:** `Device(config-if-vrrp)# shutdown`	(Optional) Disables VRRP configuration for the VRRP group. • VRRP configuration is enabled for a VRRP group by default.
Step 15	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring the Delay Period Before FHRP Client Initialization

To configure the delay period before the initialization of all FHRP clients on an interface, perform the following task:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **fhrp version vrrp v3**
4. **interface** *type number*
5. **fhrp delay** {[**minimum**] [**reload**] *seconds*}
6. **end**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **fhrp version vrrp v3**

Example:

```
Device(config)# fhrp version vrrp v3
```

Enables the ability to configure VRRPv3 and VRRS.

Note When VRRPv3 is in use, VRRPv2 is unavailable.

Step 4 **interface** *type number*

Example:

```
Device(config)# interface GigabitEthernet 0/0/0
```

Enters interface configuration mode.

Step 5 **fhrp delay** {[**minimum**] [**reload**] *seconds*}

Example:

```
Device(config-if)# fhrp delay minimum 5
```

Specifies the delay period for the initialization of FHRP clients after an interface comes up.

- The range is 0-3600 seconds.

Step 6 **end**

Example:

```
Device(config)# end
```

Returns to privileged EXEC mode.

Configuration Examples for VRRPv3 Protocol Support

Example: Enabling VRRPv3 on a Device

The following example shows how to enable VRRPv3 on a device:

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config-if-vrrp)# end
```

Example: Creating and Customizing an IPv4 VRRP Group

The following example shows how to create and customize an IPv4 VRRP group:

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config)# interface gigabitethernet0/0
Device(config-if)# vrrp 3 address-family ipv4
Device(config-if-vrrp)# address 10.0.0.1 primary
Device(config-if-vrrp)# description group 3
Device(config-if-vrrp)# match-address
Device(config-if-vrrp)# preempt delay minimum 30
Device(config-if-vrrp)# end
```

Note In the above example, the **fhrp version vrrp v3** command is used in the global configuration mode.

Example: Creating and Customizing an IPv6 VRRP Group

The following example shows how to create and customize an IPv6 VRRP group:

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config)# interface gigabitethernet0/0/0
Device(config-if)# vrrp 3 address-family ipv6
Device(config-if-vrrp)# address FE80::46B6:BEFF:FE50:DBB0 primary
```

```
Device(config-if-vrrp)# address 2001:1234::FFFF/64
Device(config-if-vrrp)# exit-vrrp
Device(config-if)# end
```

Note In above example, primary link-local address should match on all VRRP peers and the **fhrp version vrrp v3** is used in the global configuration mode

Example: Configuring the Delay Period Before FHRP Client Initialization

The following example shows how to configure the delay period before FHRP client initialization :

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config)# interface gigabitethernet0/0
Device(config-if)# fhrp delay minimum 5
Device(config-if-vrrp)# end
```

Note In the above example, a five-second delay period is specified for the initialization of FHRP clients after the interface comes up. You can specify a delay period between 0 and 3600 seconds.

Example: VRRP Status, Configuration, and Statistics Details

The following is a sample output of the status, configuration and statistics details for a VRRP group:

```
Device# show vrrp detail

GigabitEthernet0/0/0 - Group 3 - Address-Family IPv4
  State is MASTER
  State duration 1 mins 23.203 secs
  Virtual IP address is 10.0.0.254
  Virtual MAC address is 0000.5E00.0103
  Advertisement interval is 1000 msec
  Preemption enabled
  Priority is 110
  State change reason is VRRP_PRIORITY
  Master Router is 10.0.0.1 (local), priority is 110
  Master Advertisement interval is 1000 msec (expires in 598 msec)
  Master Down interval is unknown
  FLAGS: 1/1
  VRRPv3 Advertisements: sent 1985 (errors 0) - rcvd 70
  VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
  Group Discarded Packets: 3
    VRRPv2 incompatibility: 0
    IP Address Owner conflicts: 0
    Invalid address count: 0
    IP address configuration mismatch : 0
    Invalid Advert Interval: 0
    Adverts received in Init state: 3
```

```
    Invalid group other reason: 0
  Group State transition:
    Init to master: 0
    Init to backup: 3 (Last change Fri Jul 26 15:10:23.977)
    Backup to master: 2 (Last change Fri Jul 26 15:10:27.547)
    Master to backup: 0
    Master to init: 1 (Last change Fri Jul 26 15:09:28.947)
    Backup to init: 1 (Last change Fri Jul 26 14:40:55.710)

GigabitEthernet0/0/0 - Group 3 - Address-Family IPv6
  State is MASTER
  State duration 1 mins 26.772 secs
  Virtual IP address is FE80::46B6:BEFF:FE50:DBB0
  Virtual secondary IP addresses:
    2001:1234::FFFF/64
  Virtual MAC address is 0000.5E00.0203
  Advertisement interval is 1000 msec
  Preemption enabled
  Priority is 255 (owner mode)
  State change reason is VRRP_PRIORITY
  Master Router is FE80::46B6:BEFF:FE50:DBB0 (local), priority is 255
  Master Advertisement interval is 1000 msec (expires in 600 msec)
  Master Down interval is unknown
  FLAGS: 1/1
  VRRPv3 Advertisements: sent 1181 (errors 0) - rcvd 1
  VRRPv2 Advertisements: sent 0 (errors 0) - rcvd 0
  Group Discarded Packets: 0
    VRRPv2 incompatibility: 0
    IP Address Owner conflicts: 0
    Invalid address count: 0
    IP address configuration mismatch : 1
    Invalid Advert Interval: 0
    Adverts received in Init state: 0
    Invalid group other reason: 0
  Group State transition:
    Init to master: 2 (Last change Fri Jul 26 15:10:23.979)
    Init to backup: 0
    Backup to master: 0
    Master to backup: 0
    Master to init: 1 (Last change Fri Jul 26 15:09:28.947)
    Backup to init: 0

Device# exit
```

Additional References for VRRPv3 Protocol Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
FHRP commands	First Hop Redundancy Protocols Command Reference
Configuring VRRPv2	*Configuring VRRP*

Standards and RFCs

Standard/RFC	Title
RFC5798	*Virtual Router Redundancy Protocol*
RFC 6527	*Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3)*

MIBs

MIB	MIBs Link
VRRPv3 MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VRRPv3 Protocol Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Glossary

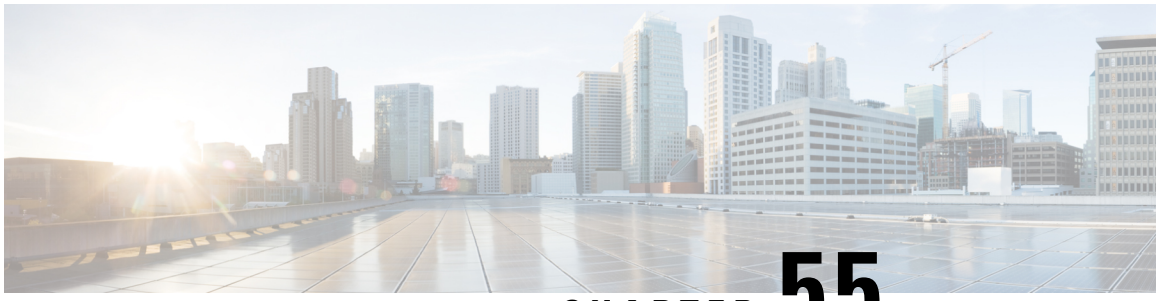
Virtual IP address owner—The VRRP router that owns the IP address of the virtual router. The owner is the router that has the virtual router address as its physical interface address.

Virtual router—One or more VRRP routers that form a group. The virtual router acts as the default gateway router for LAN clients. The virtual router is also known as a VRRP group.

Virtual router backup—One or more VRRP routers that are available to assume the role of forwarding packets if the virtual primary router fails.

Virtual primaryrouter —The VRRP router that is currently responsible for forwarding packets sent to the IP addresses of the virtual router. Usually, the virtual primary router also functions as the IP address owner.

VRRP router—A router that is running VRRP.



CHAPTER 55

VRRPv3: Object Tracking Integration

Virtual Router Redundancy Protocol (VRRP) enables a group of devices to form a single virtual device to provide redundancy. The LAN clients then can be configured with the virtual device as the default gateway. The virtual device, representing a group of devices, is also known as a VRRP group. The VRRPv3: Object Tracking Integration feature allows you to track the behavior of an object and receive notifications of changes. This module explains how object tracking, in particular the tracking of IPv6 objects, is integrated into VRRP version 3 (VRRPv3) and describes how to track an IPv6 object using a VRRPv3 group. See the "VRRP Object Tracking" section for more information on object tracking.

Information About VRRPv3: Object Tracking Integration

VRRP Object Tracking

Object tracking is an independent process that manages creating, monitoring, and removing tracked objects such as the state of the line protocol of an interface. Clients such as the Hot Standby Router Protocol (HSRP), Gateway Load Balancing Protocol (GLBP), and VRRP register their interest with specific tracked objects and act when the state of an object changes.

Each tracked object is identified by a unique number that is specified on the tracking CLI. Client processes such as VRRP use this number to track a specific object.

The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

VRRP object tracking gives VRRP access to all the objects available through the tracking process. The tracking process allows you to track individual objects such as a the state of an interface line protocol, state of an IP route, or the reachability of a route.

VRRP provides an interface to the tracking process. Each VRRP group can track multiple objects that may affect the priority of the VRRP device. You specify the object number to be tracked and VRRP is notified of

any change to the object. VRRP increments (or decrements) the priority of the virtual device based on the state of the object being tracked.

How VRRP Object Tracking Affects the Priority of a Device

The priority of a device can change dynamically if it has been configured for object tracking and the object that is being tracked goes down. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to VRRP, either immediately or after a specified delay. The object values are reported as either up or down. Examples of objects that can be tracked are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, the VRRP priority is reduced. The VRRP device with the higher priority can now become the virtual primary device if it has the **vrrp preempt** command configured. See the "VRRP Object Tracking" section for more information on object tracking.

How to Configure VRRPv3: Object Tracking Integration

Tracking an IPv6 Object using VRRPv3

SUMMARY STEPS

1. **fhrp version vrrp v3**
2. **interface** *type number*
3. **vrrp** *group-id* **address-family ipv6**
4. **track** *object-number* **decrement** *number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**fhrp version vrrp v3** **Example:** `Device(config)# fhrp version vrrp v3`	Enables you to configure Virtual Router Redundancy Protocol version 3 (VRRPv3) and Virtual Router Redundancy Service (VRRS) on a device. **Note** When VRRPv3 is in use, VRRPv2 is unavailable.
Step 2	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies an interface and enters interface configuration mode.
Step 3	**vrrp** *group-id* **address-family ipv6** **Example:** `Device(config-if)# vrrp 1 address-family ipv6`	Creates a VRRP group for IPv6 and enters VRRP configuration mode.

	Command or Action	Purpose
Step 4	**track** *object-number* **decrement** *number* **Example:** `Device(config-if-vrrp)# track 1 decrement 20`	Configures the tracking process to track the state of the IPv6 object using the VRRPv3 group. VRRP on Ethernet interface 0/0 then registers with the tracking process to be informed of any changes to the IPv6 object on the VRRPv3 group. If the IPv6 object state on serial interface VRRPv3 goes down, then the priority of the VRRP group is reduced by 20.
Step 5	**end** **Example:** `Device(config-if-vrrp)# end`	Returns to privileged EXEC mode.

Configuration Examples for VRRPv3: Object Tracking Integration

Example: Tracking an IPv6 Object using VRRPv3

In the following example, the tracking process is configured to track the state of the IPv6 object using the VRRPv3 group. VRRP on GigabitEthernet interface 0/0/0 then registers with the tracking process to be informed of any changes to the IPv6 object on the VRRPv3 group. If the IPv6 object state on serial interface VRRPv3 goes down, then the priority of the VRRP group is reduced by 20:

```
Device(config)# fhrp version vrrp v3
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# vrrp 1 address-family ipv6
Device(config-if-vrrp)# track 1 decrement 20
```

Example: Verifying VRRP IPv6 Object Tracking

```
Device# show vrrp

Ethernet0/0 - Group 1 - Address-Family IPv4
  State is BACKUP
  State duration 1 mins 41.856 secs
  Virtual IP address is 172.24.1.253
  Virtual MAC address is 0000.5E00.0101
  Advertisement interval is 1000 msec
  Preemption enabled
  Priority is 80 (configured 100)
    Track object 1 state Down decrement 20
  Master Router is 172.24.1.2, priority is 100
  Master Advertisement interval is 1000 msec (learned)
  Master Down interval is 3609 msec (expires in 3297 msec)

Device# show track ipv6 route brief

Track Type        Instance                  Parameter        State Last Change
601   ipv6 route  3172::1/32                metric threshold Down  00:08:55
```

```
602   ipv6 route  3192:ABCD::1/64           metric threshold Down  00:08:55
603   ipv6 route  3108:ABCD::CDEF:1/96      metric threshold Down  00:08:55
604   ipv6 route  3162::EF01/16             metric threshold Down  00:08:55
605   ipv6 route  3289::2/64                metric threshold Down  00:08:55
606   ipv6 route  3888::1200/64             metric threshold Down  00:08:55
607   ipv6 route  7001::AAAA/64             metric threshold Down  00:08:55
608   ipv6 route  9999::BBBB/64             metric threshold Down  00:08:55
611   ipv6 route  1111::1111/64             reachability     Down  00:08:55
612   ipv6 route  2222:3333::4444/64        reachability     Down  00:08:55
613   ipv6 route  5555::5555/64             reachability     Down  00:08:55
614   ipv6 route  3192::1/128               reachability     Down  00:08:55
```

Additional References for VRRPv3: Object Tracking Integration

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
HSRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS First Hop Redundancy Protocols Command Reference*
Troubleshooting HSRP	*Hot Standby Router Protocol: Frequently Asked Questions*

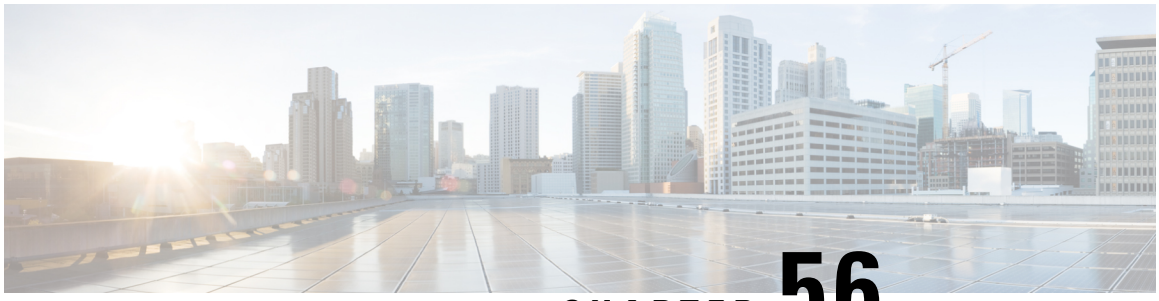
RFCs

RFCs	Title
RFC 792	*Internet Control Message Protocol*
RFC 1828	*IP Authentication Using Keyed MD5*
RFC 5798	*Virtual Router Redundancy Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VRRPv3: Object Tracking Integration



CHAPTER 56

Virtual Router Redundancy Service

Virtual Router Redundancy Service (VRRS) provides a multiclient information abstraction and management service between the Virtual Router Redundancy Protocol (VRRP), VRRS pathways and optional VRRS clients. The VRRS multiclient service provides a consistent interface with VRRP by abstracting over several First Hop Redundancy Protocols (FHRPs) and providing an idealized view of their state. VRRS manages data updates, allowing interested clients to register in one place and receive updates for named VRRP groups.

VRRP acts as a server that pushes VRRP status information out to VRRS pathways, and all registered VRRS clients. Pathways and clients obtain status on all essential information provided by VRRP, including current and previous redundancy states, active and inactive Layer 2 and Layer 3 addresses, and, in some cases, information about other redundant gateways in the network. Pathways use this information in order to provide scaled first-hop gateway redundancy across scaled interface environments. VRRS clients will also use this information to provide stateless and stateful redundancy information to clients and protocols.

Note In this module, VRRP and VRRPv3 are used interchangeably.

Restrictions for VRRS

- VRRS plug-ins must be configured on subinterfaces that are not configured with VRRP, but which share a physical interface with a VRRP group it is following.
- VRRP Version 2 (VRRPv2) is configurable only on Gigabit Ethernet interfaces.
- VRRS is currently only available for use with VRRP Version 3 (VRRPv3).

Information About VRRS

VRRS Overview

VRRS improves the scalability of VRRP. VRRS provides a stateless redundancy service to VRRS pathways and applications (VRRS clients) by monitoring VRRP. VRRS provides a database of the current VRRP state and provides a “push” data service to the VRRS pathways and clients with which it communicates. VRRP acts as a VRRS server. VRRS clients are other Cisco processes or applications that use VRRP to provide or withhold a service or resource dependent upon the state of the group. VRRS pathways are special VRRS clients that use the VRRS database information in order to provide scaled first–hop gateway redundancy across scaled interface environments.

The VRRS by itself is limited to maintaining its own state. Linking a VRRS client to a VRRP group provides a mechanism that allows VRRS to provide a service to client applications so that they can implement stateless or stateful failover. Stateless failover is failover without syncing of state. Stateful failover requires communication with a nominated backup before failure so that operational data is not lost when failover occurs.

VRRS pathways operate in a similar way to clients, but are integrated with the VRRS architecture. They provide a means to scale first–hop gateway redundancy by allowing the user the opportunity to configure a virtual address across hundreds of interfaces. The “virtual gateway” state of a VRRS pathway follows the state of an FHRP VRRS server.

Using VRRS with VRRP

VRRP provides server support for VRRS. The VRRP server pushes state and status information to VRRS when an internal update occurs. VRRS updates its internal database upon receiving a server update, and then sends push notifications to each of the VRRS clients associated with the shared name. Clients are interested in the protocol state, virtual MAC (vMAC) address, and virtual IP address information associated with a group. The association name between a client and a VRRP group is a character name string. The information provided by VRRS allows clients to perform various activities that are dependent on the state of the associated VRRP group.

VRRP notifies VRRS of its current state (primary, backup, or nonoperational initial state [INIT]). The VRRP state is then passed on to pathways or clients. A VRRP group should be configured with a name to activate VRRS. Pathways or clients should be configured with the same name to bind them with VRRS.

The VRRP group name associates the VRRP group with any clients that are configured as part of VRRS with the same name.

VRRS Servers and Clients

VRRP acts as the VRRS server. Pathways and clients act on the VRRP server state. When a VRRP group changes state, VRRS pathways and clients act by altering their behaviour (performing tasks such as shutting down interfaces or appending accounting logs) depending on the state received from VRRS.

VRRS Pathways and Pathway Manager

VRRS Pathways

A VRRS pathway is defined as an entity that will provide IPv4 or IPv6 traffic forwarding duties using the following features on an Ethernet interface (such as a physical interface, subinterface, or a Switch Virtual Interface [SVI]):

- vMAC address insertion and removal into the hardware driver using MACdb.
- Virtual IP (vIP) insertion and removal using the IPv4 and IPv6 APIs.
- Provision to associate the vIP with the interface burned-in address (BIA) MAC.
- Provision to associate the vMAC address with the interface–owned vIP.
- Maintain the association of a vMAC with a vIP on a LAN using the Address Resolution Protocol (ARP) or Neighbor Discovery Protocol.
- Maintain the switching cache (content-addressable memory or [CAM]) of connected Layer 2 devices on the LAN.
- Checkpoints all data and the pathway state with a High Availability module.

A Pathway will provide some of the above features using its association with either the VRRS Pathway L2 Controller or the VRRS Pathway L3 Controller.

VRRS Pathway Manager

The VRRS Pathway Manager provides the following features:

- Creates an association between one or more VRRS pathway instances and a single VRRS database name entry.
- Pushes configuration and state information to associated registered pathways in response to a push from VRRS.
- Provides debugging and show output to the user. The output is related to the state and configuration of the VRRS pathway manager.
- Is Online Insertion and Removal (OIR)–aware and manages pathways that may be affected by OIR events.
- Is Virtual Routing and Forwarding (VRF)–aware and manages pathways that may be affected by VRF events.

How to Configure VRRS

Configuring VRRPv3 Control Groups

Perform the following task to configure a VRRP control group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **fhrp version vrrp v3**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **vrrp** *group-id* **address-family** {**ipv4** | **ipv6**}
7. **address** *ip-address* [**primary** | **secondary**]
8. **vrrs leader** *vrrs-leader-name*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**fhrp version vrrp v3** **Example:** `Device(config)# fhrp version vrrp v3`	Enables the ability to configure VRRPv3 and VRRS. **Note** When VRRPv3 is in use, VRRPv2 is unavailable.
Step 4	**interface** *type number* **Example:** `Device(config)# interface vlan 40`	Enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.200.230 255.255.255.224`	Configures the IP address on the interface.
Step 6	**vrrp** *group-id* **address-family** {**ipv4**	**ipv6**} **Example:** `Device(config-if)# vrrp 1 address-family ipv4`	Creates a VRRP group and enters VRRP configuration mode.

	Command or Action	Purpose
Step 7	**address** *ip-address* [**primary** | **secondary**] **Example:** `Device(config-if-vrrp)# address 209.165.202.141`	Specifies a primary or secondary address for the VRRP group.
Step 8	**vrrs leader** *vrrs-leader-name* **Example:** `Device(config-if-vrrp)# vrrs leader group1`	Specifies a leader's name to be registered with VRRS and enables a VRRP group to control a VRRS pathway. • It is possible for a single VRRP instance to control more than one VRRS group. A registered VRRS name is unavailable by default.
Step 9	**end** **Example:** `Device(config-if-vrrp)# end`	Returns to privileged EXEC mode.

Configuring VRRS Pathways

Perform the following task to configure a VRRP pathway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **fhrp version vrrp v3**
4. **interface** *type number*
5. **ip address** *ip-address mask*
6. **vrrs pathway** *vrrs-leader-name*
7. **mac address** *mac-address*
8. **address** *ip-address*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**fhrp version vrrp v3** **Example:** `Device(config)# fhrp version vrrp v3`	Enables the ability to configure VRRPv3 and VRRS. **Note** When VRRPv3 is in use, VRRPv2 is unavailable.
Step 4	**interface** *type number* **Example:** `Device(config)# interface vlan 42`	Enters interface configuration mode.
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 209.165.201.25 255.255.255.224`	Configures the IP address on the interface.
Step 6	**vrrs pathway** *vrrs-leader-name* **Example:** `Device(config-if)# vrrs pathway group1`	Defines the VRRS pathway for a VRRS group and enters VRRS pathway configuration mode.
Step 7	**mac address** *mac-address* **Example:** `Device(config-if-vrrs-pw)# mac address fe24.fe24.fe24`	Specifies a MAC address used by a pathway.
Step 8	**address** *ip-address* **Example:** `Device(config-if-vrrs-pw)# address 209.165.201.10`	Defines the virtual IP for a pathway. • **Note** A VRRP group is capable of controlling more than one pathway.
Step 9	**end** **Example:** `Device(config-if-vrrs-pw)# end`	Returns to privileged EXEC mode. • **Note** Repeat steps 1 to 9 to configure more pathways.

Verifying VRRS

Perform this task to verify VRRS functions.

Note The **show** commands are not in any specific order. The **show vrrs pathway** command for different pathway states (active, inactive, and "not ready") is displayed below.

SUMMARY STEPS

1. **enable**
2. **show vrrs pathway**
3. **show vrrs pathway**
4. **show vrrs pathway**
5. **show vrrs server**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 show vrrs pathway

Displays VRRS pathway information for an active pathway with the tag name “group1” and VRRP in primary state on the VLAN interface.

Example:

```
Device# show vrrs pathway

Pathway ["group1"@Vlan42]
State is ACTIVE [VRRS push "ACTIVE"]
Virtual MAC is fe24.fe24.fe24 [Active] (0)
Address-family is v4
Options: Default Pathway=0, Owner Mode=0, Accept-Mode=1, Configured vMAC=1
Evaluation: No Shut=1, Connected=1, OIR=1, L2 Ready=1, L3 Ready=1, vMAC Ready=1,
vIP Ready=1
Virtual Address List: 209.165.201.10
```

Step 3 show vrrs pathway

Displays VRRS pathway information for an inactive pathway with the tag name “group1” and VRRP in backup state on the Ethernet 0/1 interface.

Example:

```
Device# show vrrs pathway

Pathway ["group1"@Et0/1]
State is INACTIVE [VRRS push "BACKUP"]
Virtual MAC is 0101.0101.0101 [Reserved] (0)
Address-family is v4
Options: Default Pathway=0, Owner Mode=0, Accept-Mode=1, Configured vMAC=1
Evaluation: No Shut=1, Connected=1, OIR=1, L2 Ready=1, L3 Ready=1, vMAC Ready=1,
vIP Ready=1
Virtual Address List: 209.165.201.10
```

Step 4 show vrrs pathway

Displays VRRS pathway information for a “not ready” pathway with the tag name “group1” and VRRP in backup state on the Ethernet 0/1 interface.

Example:

```
Device# show vrrs pathway

Pathway ["group1"@Et0/1]
State is NOT READY [VRRS push "INIT"]
Virtual MAC is 0101.0101.0101 [Reserved] (0)
Address-family is v4
Options: Default Pathway=0, Owner Mode=0, Accept-Mode=1, Configured vMAC=1
Evaluation: No Shut=1, Connected=1, OIR=1, L2 Ready=1, L3 Ready=1, vMAC Ready=1,
vIP Ready=1
Virtual Address List: 209.165.201.10
```

Step 5 **show vrrs server**

Displays VRRS server information.

Example:

```
Device# show vrrs pathway

Pathway ["group1"@Et0/1]
State is INACTIVE [VRRS push "BACKUP"]
Virtual MAC is 0101.0101.0101 [Reserved] (0)
Address-family is v4
Options: Default Pathway=0, Owner Mode=0, Accept-Mode=1, Configured vMAC=1
Evaluation: No Shut=1, Connected=1, OIR=1, L2 Ready=1, L3 Ready=1, vMAC Ready=1,
vIP Ready=1
Virtual Address List: 209.165.201.10
```

The table below describes significant fields in the sample output:

Field	Description
State	Current state of VRRS on an interface. The values displayed are “ACTIVE”, “INACTIVE”, “NOT READY”, or “BACKUP”.
Virtual MAC	Virtual MAC address that is reserved for an interface.
Address-family	IPv4 or IPv6 address family.
Default Pathway	Indicates that the pathway has been implicitly created from a VRRP group, if the value is 1. If the value is 0, it indicates that the pathway has been explicitly created using the **vrrs pathway** command.
Owner Mode	Indicates that the interface IP address is specified if the value is 1.
Accept-Mode	Indicates that traffic to a particular virtual IP address is accepted if the value is 1.

Field	Description
Configured vMAC	Indicates that a virtual MAC address is configured if the value is 1.
No Shut	Indicates that the interface has been set to no shutdown mode if the value is 1.
Connected	Indicates that the VRRS pathway is connected to a VRRS group, if the value is 1.
OIR	Indicates online insertion and removal (OIR) of interface line cards on a device is complete if the value is 1.
L2 Ready	Indicates that the Layer 2 interface is up if the value is 1.
L3 Ready	Indicates that the Layer 3 interface is up if the value is 1.
vMAC Ready	Indicates that the virtual MAC address has been assigned to an interface if the value is 1.
vIP Ready	Indicates that the virtual IP address has been assigned to an interface if the value is 1.
Virtual Address List	Address list of the virtual IPv4 or IPv6 addresses.
Interface	Name of the interface where the pathway is defined.
vMAC	Virtual MAC address that is assigned to an interface.
vIP Address	Virtual IP address that is assigned to an interface.
Tags Connected	The specific tag name that is currently connected to a pathway on an interface.

Configuration Examples for VRRS

Example: Configuring VRRPv3 Control Groups

The following example shows how to configure a VRRPv3 control group:

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config)# interface vlan 40
Device(config-if)# ip address 209.165.200.230 255.255.255.224
Device(config-if)# vrrp 1 address-family ipv4
Device(config-if-vrrp)# address 209.165.202.141
Device(config-if-vrrp)# vrrs leader group1
```

```
Device(config-if-vrrp)# end
```

Note In the above example, the **fhrp version vrrp v3** command is used in global configuration mode.

Example: Configuring VRRS pathways

The following example shows how to configure a VRRS pathway:

```
Device> enable
Device# configure terminal
Device(config)# fhrp version vrrp v3
Device(config)# interface vlan 42
Device(config-if)# ip address 209.165.201.25 255.255.255.224
Device(config-if)# vrrs pathway group1
Device(config-if-vrrs-pw)# mac address fe24.fe24.fe24
Device(config-if-vrrs-pw)# address 209.165.201.10
Device(config-if-vrrs-pw)# end
```

Note In the above example, the **fhrp version vrrp v3** command is used in global configuration mode.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
FHRP commands	First Hop Redundancy Protocols Command Reference
Configuring VRRPv2	"Configuring VRRP" module in the *First Hop Redundancy Protocols Configuration Guide*
VRRPv3 Protocol Support	"VRRPv3 Protocol Support" module in the *First Hop Redundancy Protocols Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC5798	*Virtual Router Redundancy Protocol*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Virtual Router Redundancy Service

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.