Advanced RIP Features

The Advanced RIP Features contained in this configuration module cover the implementation of RFC 1724, which allows you to monitor RIPv2 using SNMP, and the information about configuring the cable modem HFC RIP Relay feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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.

Information About Advanced RIP Features

Cable HFC

Cable technology has been adapting to the deployment of fiber since 1994, leading to hybrid solutions known as hybrid fiber-coaxial (HFC). HFC networks contain both optical-fiber and coaxial cable lines. Optical fiber is deployed from the cable headend to cable operator subscribers with up to 2000 subscribers. Coaxial cable

is deployed from the optical-fiber feeders to each subscriber. Hybrid networks provide the bandwidth and reliability of optical fiber at a lower cost than a pure fiber network.

HFC RIP Relay

The cable modem HFC RIP Relay feature allows the delivery of Routing Information Protocol (RIP) messages from a Cisco IOS router containing an integrated cable modem to the hybrid fiber-coaxial (HFC) cable modem termination system (CMTS) when they are on different subnets. The integrated cable modem may be physically integrated into the router or via a cable modem high-speed WAN interface card (HWIC). In previous Cisco IOS releases, RIP messages were rejected by the CMTS because the interface on the Cisco IOS router was in a different subnet from the CMTS. The solution involves trapping and handling RIP messages by the cable modem and ensuring that the RIP messages are forwarded to the router. The cable modem HFC RIP Relay feature enhances the scalability, security, and certification requirements of cable operators who require RIP to provision and manage customer cable modems.

In the provisioning systems used by some cable operators, when a Cisco IOS router containing an integrated cable modem is connected to a CMTS, RIP messages are rejected because the IP address derived from a DHCP request for the router is from a different pool of IP addresses than for the cable modems. The RIP messages are rejected by the CMTS because the interface on the Cisco IOS router is in a different subnet from the CMTS. Without requiring additional configuration on the CMTS, the HFC RIP Relay feature enables the cable modem to bridge the RIP messages between the Cisco IOS router and the CMTS.

The cable modem HFC RIP Relay feature is implemented in Cisco IOS Release 12.4(15)XY, 12.4(20)T, and later releases. The feature requires the cable modem firmware version filename of C21031014bFU07192007.CDF in the United States or the cable modem firmware version filename of C21041014bFU07192007.CDF in Europe and Japan, and the feature is turned off by default. To enable HFC RIP relay, use the new **service-module ip rip relay** command-line interface (CLI) command.

Support is added for configuring a static IP address on the cable modem interface. Configuring a static IP address for the Cisco IOS router with an integrated cable modem is also supported in Cisco IOS Release 12.4(15)XY, 12.4(20)T, and later releases using the **ip address** command.

Benefits of the RIPv2 MIB

The RFC 1724 RIPv2 MIB extensions allow network managers to monitor the RIPv2 routing protocol using SNMP through the addition of new global counters and table objects that previously were not supported by the RFC 1389 RIPv2 MIB. The new global counters and table objects are intended to facilitate quickly changing routes or failing neighbors.

RIPv2 MIB

This document describes the Cisco IOS implementation of RFC 1724, *RIP Version 2 MIB Extensions* . RIPv2 using Simple Network Management Protocol (SNMP).

This section describes the MIB objects that are provided by RFC 1724 definitions. The RIPv2 MIB consists of the following managed objects:

- Global counters--Used to keep track of changing routes or neighbor changes.
- Interface status table--Defines objects that are used to keep track of statistics specific to interfaces.

- Interface configuration table--Defines objects that are used to keep track of interface configuration statistics.
- Peer table--Defined to monitor neighbor relationships. This object is not implemented in Cisco IOS software.

The tables below show the objects that are provided by RFC 1724 RIPv2 MIB definitions. The objects are listed in the order in which they appear within the RFC 1724 RIPv2 MIB, per the tables that describe them. The statistics for all of the objects in the global counters can be obtained by querying the rip2Globals object identifier (OID) using **snmpwalk** or a similar SNMP toolset command on your Network Management Station (NMS).

The table below shows the RFC 1724 RIPv2 MIB global counter objects.

Table 1: RFC 1724 RIPv2 MIB Global Counters Objects

Global Counter	Object	Description
rip2Globals	rip2GlobalRouteChanges	Number of route changes made to the IP route database by RIP. The number is incremented when a route is modified.
	rip2GlobalQueries	Number of responses sent to RIP queries from other systems. The number is incremented when RIP responds to a query from another system.

The objects in the RFC 1724 RIPv2 MIB interface table track information on a per-interface basis. All objects in the RFC 1724 RIPv2 MIB interface table, except for the rip2IfStatAddress object, represent newly tracked data within RIP. There are no equivalent **show** commands for these objects. All objects in the RIPv2 MIB interface table are implemented read-only.

The table below shows the RFC 1724 RIPv2 MIB interface table objects. The statistics for all objects in the interface table can be obtained by querying the sequence name Rip2IfStatEntry using **snmpwalk** or a similar SNMP toolset command on your NMS.

Table 2: RFC 1724 RIPv2 MIB Interface Table Objects

Sequence Name	Object	Description
Rip2IfStatEntry	rip2IfStatAddress	The IP address of this system on the indicated subnet. For unnumbered interfaces, the value of 0.0.0.N, where the least significant 24 bits (N) are the ifIndex for the IP interface in network byte order.
	rip2IfStatRcvBadPackets	The number of RIP response packets received by the RIP process that were subsequently discarded for any reason. For example, a version 0 packet or an unknown command type.
	rip2IfStatRcvBadRoutes	The number of routes, in valid RIP packets, that were ignored for any reason. This is incremented when: • The address family identifier does not equal AF_INET. • A RIP v2 update is received and the address is not a RIP multicast address (244.0.0.0) or RIP broadcast address (255.255.255.255). • A RIP v2 update is received and the address is a martian address.
	rip2IfStatSentUpdates	The number of triggered RIP updates actually sent on this interface. This explicitly does not include full updates sent containing new information.
	rip2IfStatStatus	This value is always set to 1.

The objects in the RFC 1724 RIPv2 MIB interface configuration table track information on a per- interface basis. Except for the Rip2IfConfAuthType object, the data for the objects in the RFC 1724 RIPv2 MIB interface configuration table can also be gathered using the **show ip protocol** commands. All objects in the RIPv2 MIB interface table are implemented read-only.

The table below shows the RIPv2 MIB interface configuration table objects. The statistics for all objects in the configuration table can be obtained by querying the sequence name rip2IfConfEntry using **snmpwalk** or a similar SNMP toolset command on your NMS.

Table 3: RFC 1724 RIPv2 MIB Interface Configuration Table Object Types

Sequence Name	Object Type	Description
rip2IfConfEntry	rip2IfConfAddress	The IP address of this system on the indicated subnet. For unnumbered interfaces, the value 0.0.0.N, where the least significant 24 bits (N) are the ifIndex for the IP interface in network byte order.
	rip2IfConfDomain	This value is always equal to "".
	rip2IfConfAuthType	The type of authentication used on this interface.
	rip2IfConfAuthKey	The value to be used as the authentication key whenever the corresponding instance of rip2IfConfAuthType has a value other than no authentication.
	rip2IfConfSend	The version of RIP updates that are sent on this interface.
	rip2IfConfReceive	The version of RIP updates that are accepted on this interface.
	rip2IfConfDefaultMetric	This variable indicates the metric that is used for the default route entry in RIP updates originated on this interface.
	rip2IfConfStatus	This value is always set to 1.
	rip2IfConfSrcAddress	The IP address that this system will use as a source address on this interface. If it is a numbered interface, this must be the same value as rip2IfConfAddress. On unnumbered interfaces, it must be the value of rip2IfConfAddress for some interface on the system.

SNMP Community Strings

Routers can have multiple read-only SNMP community strings. When you configure an SNMP read-only community string for the **snmp-server** command on the router, an existing SNMP **snmp-server** read-only community string is not overwritten. For example, if you enter the **snmp-server community** *string1* **ro** and **snmp-server community** *string2* **ro** commands on the router, the router will have two valid read-only community strings--*string1* and *string2*. If this is not the behavior that you desire, use the **no snmp-server community** *string* **ro** command to remove an existing SNMP read-only community string.

Timesaver If you already have an SNMP read-only community string configured on your router you do not need to perform this task. After you load Cisco IOS Release 12.4(6)T or a later release on your router, you can use SNMP commands on your NMS to query the RFC 1724 RIPv2 MIB on your router

How to Configure Advanced RIP Features

Configuring HFC RIP Relay

This section contains the following tasks:

Prerequisites

The HFC RIP Relay feature requires an Integrated Services Router (ISR) with an integrated cable modem and Cisco IOS Release 12.4(15)XY, 12.4(20)T, or later release and one of the following:

- Cable modem firmware version filename of C21031014bFU07192007.CDF in the United States
- Cable modem firmware version filename of C21041014bFU07192007.CDF in Europe and Japan

ISR cable products include the Cisco 815, Cisco 1805, and the cable modem HWIC in the Cisco 1800, 2800, and 3800 series routers.

Restrictions

The HFC RIP Relay feature does not support multiple cable modem HWICs in a single router.

Enabling HFC RIP Relay

Perform this task to enable RIP relay on an integrated cable modem. In this task, a static IP address is configured for the cable modem interface and RIP relay is enabled on the interface. Validation of the source IP address of incoming RIP routing updates is disabled to allow RIP updates from unknown sources. RIP is defined as the routing protocol to be used on all interfaces that are connected to networks 10.0.0.0 and 172.18.0.0.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **service-module ip rip relay**
6. **exit**
7. **router rip**
8. **version** {**1** | **2**}
9. **no validate-update-source**
10. **network** *ip-address*
11. **end**

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	**interface** *type number* **Example:** `Router(config)# interface cable-modem 0/3/0`	Configures an interface type and enters interface configuration mode. • In this example, cable-modem interface 0/3/0 is configured.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.5.5.5 255.255.255.0`	Sets a primary or secondary IP address for an interface. • In this example, the static IP address of 10.5.5.5 is configured under the cable-modem interface.
Step 5	**service-module ip rip relay** **Example:** `Router(config-if)# service-module ip rip relay`	Enables RIP relay in the ISR cable-modem driver.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode for the specified routing process. • In this example, a RIP routing process is configured.
Step 8	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies a RIP version used globally by the router. • In this example, the software sends and receives RIP version 2 packets.
Step 9	**no validate-update-source** **Example:** `Router(config-router)# no validate-update-source`	Disables the validation of the source IP address of incoming RIP routing updates. • When the validation check is enabled, the software ensures that the source IP address of incoming routing updates is on the same IP network as one of the addresses defined for the receiving interface. • In this example, the router is configured not to perform validation checks on the source IP address of incoming RIP updates.
Step 10	**network** *ip-address* **Example:** `Router(config-router)# network 10.0.0.0`	Specifies a list of networks for the RIP routing process. • In this example, RIP is defined as the routing protocol to be used on all interfaces connected to network 10.0.0.0.
Step 11	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Enabling HFC RIP Relay for a Single Subnet and Disabling Split-Horizon

Perform this task to enable RIP relay on an ISR cable modem. In this task, a static IP address is configured for the cable-modem interface and RIP relay is enabled on the interface. Split-horizon is disabled, and RIP is defined as the routing protocol to be used on all interfaces connected to network 10.0.0.0.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **service-module ip rip relay**
6. **no ip split-horizon**
7. **exit**
8. **router rip**
9. **version** {**1** | **2**}
10. **no validate-update-source**
11. **network** *ip-address*
12. **end**

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	**interface** *type number* **Example:** `Router(config)# interface cable-modem 0/3/0`	Configures an interface type and enters interface configuration mode. • In this example, cable-modem interface 0/3/0 is configured.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.5.5.5 255.255.255.0`	Sets a primary or secondary IP address for an interface. • In this example, a static IP address of 10.5.5.5 is configured under the cable-modem interface.

	Command or Action	Purpose
Step 5	**service-module ip rip relay** **Example:** `Router(config-if)# service-module ip rip relay`	Enables RIP relay in the ISR cable-modem driver.
Step 6	**no ip split-horizon** **Example:** `Router(config-if)# no ip split-horizon`	Disables split horizon, allowing routing updates to be sent from the interface over which the route was learned.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**router rip** **Example:** `Router(config)# router rip`	Enters router configuration mode for the specified routing process. • In this example, a RIP routing process is configured.
Step 9	**version** {**1** | **2**} **Example:** `Router(config-router)# version 2`	Specifies a RIP version used globally by the router. • In this example, the software sends and receives RIP version 2 packets.
Step 10	**no validate-update-source** **Example:** `Router(config-router)# no validate-update-source`	Disables the validation of the source IP address of incoming RIP routing updates. • When the validation check is enabled, the software ensures that the source IP address of incoming routing updates is on the same IP network as one of the addresses defined for the receiving interface. • In this example, the router is configured not to perform validation checks on the source IP address of incoming RIP updates.
Step 11	**network** *ip-address* **Example:** `Router(config-router)# network 10.0.0.0`	Specifies a list of networks for the RIP routing process. • In this example, RIP is defined as the routing protocol to be used on all interfaces connected to network 10.0.0.0.

	Command or Action	Purpose
Step 12	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Verifying the Configuration of HFC RIP Relay

Use the following steps to verify the local configuration of HFC RIP relay on a router.

SUMMARY STEPS

1. **enable**
2. **show running-config** [*options*]

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show running-config** [*options*]
Displays the running configuration on the local router. The output will display the configuration of the **service-module ip rip relay** command in the cable-modem interface section and of the RIP commands under the RIP protocol section.

Example:

```
Router# show running-config
.
.
.
interface Cable-Modem0/3/0
 ip address 172.20.0.2 255.255.255.0
 service-module ip rip relay
 no ip split-horizon
.
.
.
router rip
 version 2
 no validate-update-source
 network 10.0.0.0
 network 172.18.0.0
.
```

Enabling RIPv2 Monitoring with SNMP Using the RIPv2 RFC 1724 MIB Extensions

This section contains the following tasks:

Prerequisites

- RIPv2 must be configured on the router.
- Your SNMP NMS must have the RFC 1724 RIPv2 MIB installed.
- Your SNMP NMS must have the following MIBs installed because RFC 1724 imports data types and object identifiers (OIDs) from them:
  - SNMPv2-SMI
  - SNMPv2-TC
  - SNMPv2-CONF
  - RFC1213-MIB

Restrictions

This implementation of the RIPv2 MIB does not track any data associated with a RIP Virtual Routing and Forwarding (VRF) instance. Only interfaces that are assigned IP addresses in the IP address space configured by the **network** command in RIP router configuration mode are tracked. Global data is tracked only for changes to the main routing table.

Enabling SNMP Read-Only Access on the Router

There are no router configuration tasks required for the RIPv2: RFC 1724 MIB Extensions feature itself. SNMP read-only access to the objects in the RFC 1724 RIPv2 MIB is enabled when you configure the SNMP server read-only community string on the router.

Note When you configure an SNMP server read-only community string on the router, you are granting SNMP read-only access to the objects that support read-only access in all MIBs that are available in the version of Cisco IOS software that is running on the router.

Perform this task to configure the SNMP server read-only community string on the router to enable SNMP read-only access to MIB objects (including the RFC 1724 RIPv2 MIB extensions) on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string1* **ro**
4. **end**

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	**snmp-server community** *string1* **ro** **Example:** `Router(config)# snmp-server community T8vCx3 ro`	Enables SNMP read-only access to the objects in the MIBs that are included in the version of Cisco IOS software that is running on the router. **Note** For security purposes, do not use the standard default value of *public* for your read-only community string. Use a combination of uppercase and lowercase letters and numbers for the password.
Step 4	**end** **Example:** `Router(config)# end`	Ends your configuration session and returns to privileged EXEC mode.

Verifying the Status of the RIPv2 RFC 1724 MIB Extensions on the Router and Your Network Management Station

Perform this optional task on your NMS to verify the status of the RFC 1724 RIPv2 MIB extensions on the router and on your NMS.

Prerequisites

Your NMS must have the RFC 1724 MIB installed.

Note This task uses the NET-SNMP toolset that is available in the public domain. The step that is documented uses a terminal session on an NMS that is running Linux. Substitute the SNMP command from the SNMP toolset on your NMS as appropriate when you perform this task.

SUMMARY STEPS

1. **snmpwalk -m all -v2c** *ip-address* **-c** *read-only-community-string* **rip2Globals**

DETAILED STEPS

snmpwalk -m all -v2c *ip-address* **-c** *read-only-community-string* **rip2Globals**
Use the **snmpwalk** command for the **rip2Globals** object in the RFC 1724 RIPv2 MIB to display the data for the objects associated with this object. This step verifies that the NMS is configured to send queries for objects in the RFC 1724 RIPv2 MIB and that the router is configured to respond to the queries.

Example:

```
$ snmpwalk -m all -v2c 10.0.0.253 -c T8vCx3 rip2Globals
RIPv2-MIB::rip2GlobalRouteChanges.0 = Counter32: 5
RIPv2-MIB::rip2GlobalQueries.0 = Counter32: 1
$
```

Configuration Examples for Advanced RIP Features

Configuration Examples for HFC RIP Relay

This section contains the following examples:

Enabling HFC RIP Relay Example

The following example enables RIP relay on an ISR cable modem. A static IP address is configured for the cable-modem interface, and RIP relay is enabled on the interface. Validation of the source IP address of incoming RIP routing updates is disabled to allow RIP updates from unknown sources. RIP is defined as the routing protocol to be used on all interfaces connected to networks 10.0.0.0 and 172.18.0.0.

```
interface Cable-Modem0/3/0
 ip address 10.5.5.5 255.255.255.0
 service-module ip rip relay
 exit
router rip
 version 2
 no validate-update-source
 network 10.0.0.0
 network 172.18.0.0
```

Enabling HFC RIP Relay for a Single Subnet and Disabling Split-Horizon Example

The following example enables RIP relay on an ISR cable modem. A static IP address is configured for the cable-modem interface, and RIP relay is enabled on the interface. Validation of the source IP address of incoming RIP routing updates is disabled to allow RIP updates from unknown sources, and split-horizon is disabled. RIP is defined as the routing protocol to be used on all interfaces connected to network 172.20.0.0.

```
interface Cable-Modem0/3/0
 ip address 172.20.0.2 255.255.255.0
 service-module ip rip relay
 no ip split-horizon
 exit
router rip
 version 2
 no validate-update-source
network 172.20.0.0
```

Configuration Examples for RIPv2 Monitoring with SNMP Using the RIPv2 RFC1724 MIB Extensions

Querying the RIP Interface Status Table Objects Example

The following example shows how to send an SNMP query to obtain data for all objects in the RIP interface status table using the **snmpwalk** command.

```
$ snmpwalk -m all -v2c 10.0.0.253 -c T8vCx3 Rip2IfStatEntry
RIPv2-MIB::rip2IfStatAddress.10.0.0.253 = IpAddress: 10.0.0.253
RIPv2-MIB::rip2IfStatAddress.172.16.1.1 = IpAddress: 172.16.1.1
RIPv2-MIB::rip2IfStatAddress.172.16.2.1 = IpAddress: 172.16.2.1
RIPv2-MIB::rip2IfStatAddress.172.17.1.1 = IpAddress: 172.17.1.1
RIPv2-MIB::rip2IfStatAddress.172.17.2.1 = IpAddress: 172.17.2.1
RIPv2-MIB::rip2IfStatRcvBadPackets.10.0.0.253 = Counter32: 0
RIPv2-MIB::rip2IfStatRcvBadPackets.172.16.1.1 = Counter32: 1654
RIPv2-MIB::rip2IfStatRcvBadPackets.172.16.2.1 = Counter32: 1652
RIPv2-MIB::rip2IfStatRcvBadPackets.172.17.1.1 = Counter32: 1648
RIPv2-MIB::rip2IfStatRcvBadPackets.172.17.2.1 = Counter32: 1649
RIPv2-MIB::rip2IfStatRcvBadRoutes.10.0.0.253 = Counter32: 0
RIPv2-MIB::rip2IfStatRcvBadRoutes.172.16.1.1 = Counter32: 0
RIPv2-MIB::rip2IfStatRcvBadRoutes.172.16.2.1 = Counter32: 0
RIPv2-MIB::rip2IfStatRcvBadRoutes.172.17.1.1 = Counter32: 0
RIPv2-MIB::rip2IfStatRcvBadRoutes.172.17.2.1 = Counter32: 0
RIPv2-MIB::rip2IfStatSentUpdates.10.0.0.253 = Counter32: 0
RIPv2-MIB::rip2IfStatSentUpdates.172.16.1.1 = Counter32: 0
RIPv2-MIB::rip2IfStatSentUpdates.172.16.2.1 = Counter32: 0
RIPv2-MIB::rip2IfStatSentUpdates.172.17.1.1 = Counter32: 0
RIPv2-MIB::rip2IfStatSentUpdates.172.17.2.1 = Counter32: 0
RIPv2-MIB::rip2IfStatStatus.10.0.0.253 = INTEGER: active(1)
RIPv2-MIB::rip2IfStatStatus.172.16.1.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfStatStatus.172.16.2.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfStatStatus.172.17.1.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfStatStatus.172.17.2.1 = INTEGER: active(1)
```

The following example shows how to send an SNMP query to obtain data for the rip2IfStatStatus object for all the interfaces in the RIP interface status table using the **snmpwalk** command.

```
$ snmpwalk -m all -v2c 10.0.0.253 -c T8vCx3 rip2IfStatStatus
RIPv2-MIB::rip2IfStatStatus.10.0.0.253 = INTEGER: active(1)
RIPv2-MIB::rip2IfStatStatus.172.16.1.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfStatStatus.172.16.2.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfStatStatus.172.17.1.1 = INTEGER: active(1)
```

```
RIPv2-MIB::rip2IfStatStatus.172.17.2.1 = INTEGER: active(1)
$
```

The following example shows how to send an SNMP query to obtain data for the rip2IfStatStatus object for a specific interface IP address in the RIP interface status table using the **snmpget** command.

```
$ snmpget -m all -v2c 10.0.0.253 -c T8vCx3 rip2IfStatStatus.10.0.0.253
RIPv2-MIB::rip2IfStatStatus.10.0.0.253 = INTEGER: active(1)
```

Querying the RIP Interface Configuration Table Objects Example

The following example shows how to send an SNMP query to obtain data for all objects in the RIP interface configuration table using the **snmpwalk** command.

```
$ snmpwalk -m all -v2c 10.0.0.253 -c T8vCx3 rip2IfConfEntry
RIPv2-MIB::rip2IfConfAddress.10.0.0.253 = IpAddress: 10.0.0.253
RIPv2-MIB::rip2IfConfAddress.172.16.1.1 = IpAddress: 172.16.1.1
RIPv2-MIB::rip2IfConfAddress.172.16.2.1 = IpAddress: 172.16.2.1
RIPv2-MIB::rip2IfConfAddress.172.17.1.1 = IpAddress: 172.17.1.1
RIPv2-MIB::rip2IfConfAddress.172.17.2.1 = IpAddress: 172.17.2.1
RIPv2-MIB::rip2IfConfDomain.10.0.0.253 = ""
RIPv2-MIB::rip2IfConfDomain.172.16.1.1 = ""
RIPv2-MIB::rip2IfConfDomain.172.16.2.1 = ""
RIPv2-MIB::rip2IfConfDomain.172.17.1.1 = ""
RIPv2-MIB::rip2IfConfDomain.172.17.2.1 = ""
RIPv2-MIB::rip2IfConfAuthType.10.0.0.253 = INTEGER: noAuthentication(1)
RIPv2-MIB::rip2IfConfAuthType.172.16.1.1 = INTEGER: noAuthentication(1)
RIPv2-MIB::rip2IfConfAuthType.172.16.2.1 = INTEGER: noAuthentication(1)
RIPv2-MIB::rip2IfConfAuthType.172.17.1.1 = INTEGER: noAuthentication(1)
RIPv2-MIB::rip2IfConfAuthType.172.17.2.1 = INTEGER: noAuthentication(1)
RIPv2-MIB::rip2IfConfAuthKey.10.0.0.253 = ""
RIPv2-MIB::rip2IfConfAuthKey.172.16.1.1 = ""
RIPv2-MIB::rip2IfConfAuthKey.172.16.2.1 = ""
RIPv2-MIB::rip2IfConfAuthKey.172.17.1.1 = ""
RIPv2-MIB::rip2IfConfAuthKey.172.17.2.1 = ""
RIPv2-MIB::rip2IfConfSend.10.0.0.253 = INTEGER: ripVersion2(4)
RIPv2-MIB::rip2IfConfSend.172.16.1.1 = INTEGER: ripVersion2(4)
RIPv2-MIB::rip2IfConfSend.172.16.2.1 = INTEGER: ripVersion2(4)
RIPv2-MIB::rip2IfConfSend.172.17.1.1 = INTEGER: ripVersion2(4)
RIPv2-MIB::rip2IfConfSend.172.17.2.1 = INTEGER: ripVersion2(4)
RIPv2-MIB::rip2IfConfReceive.10.0.0.253 = INTEGER: rip2(2)
RIPv2-MIB::rip2IfConfReceive.172.16.1.1 = INTEGER: rip2(2)
RIPv2-MIB::rip2IfConfReceive.172.16.2.1 = INTEGER: rip2(2)
RIPv2-MIB::rip2IfConfReceive.172.17.1.1 = INTEGER: rip2(2)
RIPv2-MIB::rip2IfConfReceive.172.17.2.1 = INTEGER: rip2(2)
RIPv2-MIB::rip2IfConfDefaultMetric.10.0.0.253 = INTEGER: 1
RIPv2-MIB::rip2IfConfDefaultMetric.172.16.1.1 = INTEGER: 1
RIPv2-MIB::rip2IfConfDefaultMetric.172.16.2.1 = INTEGER: 1
RIPv2-MIB::rip2IfConfDefaultMetric.172.17.1.1 = INTEGER: 1
RIPv2-MIB::rip2IfConfDefaultMetric.172.17.2.1 = INTEGER: 1
RIPv2-MIB::rip2IfConfStatus.10.0.0.253 = INTEGER: active(1)
RIPv2-MIB::rip2IfConfStatus.172.16.1.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfConfStatus.172.16.2.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfConfStatus.172.17.1.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfConfStatus.172.17.2.1 = INTEGER: active(1)
RIPv2-MIB::rip2IfConfSrcAddress.10.0.0.253 = IpAddress: 10.0.0.253
RIPv2-MIB::rip2IfConfSrcAddress.172.16.1.1 = IpAddress: 172.16.1.1
RIPv2-MIB::rip2IfConfSrcAddress.172.16.2.1 = IpAddress: 172.16.2.1
RIPv2-MIB::rip2IfConfSrcAddress.172.17.1.1 = IpAddress: 172.17.1.1
RIPv2-MIB::rip2IfConfSrcAddress.172.17.2.1 = IpAddress: 172.17.2.1
$
```

The following example shows how to send an SNMP query to obtain data for the rip2IfConfAddress object for all interfaces in the RIP interface configuration table using the **snmpwalk** command.

```
$ snmpwalk -m all -v2c 10.0.0.253 -c T8vCx3 rip2IfConfAddress
RIPv2-MIB::rip2IfConfAddress.10.0.0.253 = IpAddress: 10.0.0.253
RIPv2-MIB::rip2IfConfAddress.172.16.1.1 = IpAddress: 172.16.1.1
```

```
RIPv2-MIB::rip2IfConfAddress.172.16.2.1 = IpAddress: 172.16.2.1
RIPv2-MIB::rip2IfConfAddress.172.17.1.1 = IpAddress: 172.17.1.1
RIPv2-MIB::rip2IfConfAddress.172.17.2.1 = IpAddress: 172.17.2.1
$
```

Additional References

The following sections provide references related to advanced RIP configuration.

Related Documents

Related Topic	Document Title
Configuring RIP	"Configuring Routing Information Protocol"
Configuring protocol-independent routing features	"Configuring IP Routing Protocol-Independent Features"
Configuring Frame Relay	" Configuring Frame Relay"
Cable modem HWIC card configuration	*Cisco Cable Modem High-Speed WAN Interface Cards Configuration Guide*
RIP commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	*Cisco IOS IP Routing: RIP Command Reference*
SNMP configuration	"Configuring SNMP Support"
SNMP commands	*Cisco IOS Network Management Command Reference*

Standards

Standard	Title
No new or modified standards are supported. and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
RIPv2 MIB	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 1724	*RIP Version 2 MIB Extensions*
RFC 2082	*RIP-2 MD5 Authentication*
RFC 2453	*RIP Version 2*

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 Advanced RIP Features

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 Advanced RIP Features

Feature Name	Releases	Feature Information
HFC RIP Relay	12.4(15)XY 12.4(20)T 15.0(1)M 12.2(33)SRE	The HFC RIP Relay feature allows the delivery of Routing Information Protocol (RIP) messages from a Cisco IOS router containing a cable HWIC to the HFC CMTS when they are on different subnets. Configuring a static IP address is now also supported on a cable modem interface. The following command was introduced by this feature: **service-module ip rip relay**.
RIPv2: RFC 1724 MIB Extension	12.4(6)T 15.0(1)M 12.2(33)SRE	This feature introduces the Cisco IOS implementation of RFC 1724, *RIP Version 2 MIB Extensions* . RFC 1724 defines MIB objects that allow the management and limited control of RIPv2 using SNMP.

Glossary

OID--object identifier. A managed object within the object tree.

SNMP--Simple Network Management Protocol. Aprotocol used to monitor and manage networking devices.

snmpget--An SNMP command to query statistics from a specific OID in the MIB.