Configure RIP

- RIP, on page 1
- Prerequisites for RIP, on page 3
- Guidelines and Limitations for RIP, on page 3
- Default Settings for RIP Parameters, on page 4
- RIPs, on page 4
- Verify the RIP Configuration, on page 16
- Display the RIP Statistics, on page 17
- Configuration Examples for RIP, on page 18
- Related Topics, on page 18

RIP

RIP

RIP uses User Datagram Protocol (UDP) data packets to exchange routing information in small internetworks. RIPv2 supports IPv4. RIPv2 uses an optional authentication feature supported by the RIPv2 protocol (see the RIPv2 Authentication section).

RIP Message Types

- Request—Sent to the multicast address 224.0.0.9 to request route updates from other RIP-enabled routers.
- Response—Sent every 30 seconds by default (see the Verifying the RIP Configuration section). The router also sends response messages after it receives a request message. The response message contains the entire RIP route table. RIP sends multiple response packets for a request if the RIP routing table cannot fit in one response packet.

RIPv2 Authentication

You can configure authentication on RIP messages to prevent unauthorized or invalid routing updates in your network. Cisco NX-OS supports a simple password or an MD5 authentication digest.

You can configure the RIP authentication per interface by using keychain management for the authentication keys. Keychain management allows you to control changes to the authentication keys used by an MD5

authentication digest or simple text password authentication. See the Cisco Nexus 9000 Series NX-OS Security Configuration Guide for more details about creating keychains.

To use an MD5 authentication digest, you configure a password that is shared at the local router and all remote RIP neighbors. Cisco NX-OS creates an MD5 one-way message digest based on the message itself and the encrypted password and sends this digest with the RIP message (Request or Response). The receiving RIP neighbor validates the digest by using the same encrypted password. If the message has not changed, the calculation is identical, and the RIP message is considered valid.

An MD5 authentication digest also includes a sequence number with each RIP message to ensure that no message is replayed in the network.

Split Horizons

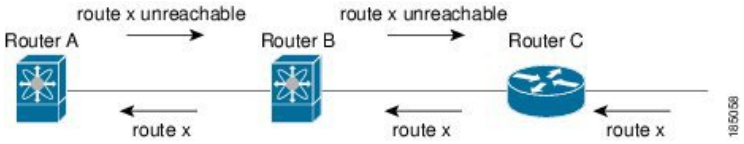
You can use split horizon to ensure that RIP never advertises a route out of the interface where it was learned.

Split horizon is a method that controls the sending of RIP update and query packets. When you enable split horizon on an interface, Cisco NX-OS does not send update packets for destinations that were learned from this interface. Controlling update packets in this manner reduces the possibility of routing loops.

You can use split horizon with poison reverse to configure an interface to advertise routes learned by RIP as unreachable over the interface that learned the routes.

The following figure shows a sample RIP network with split horizon and poison reverse enabled.

Figure 1: RIP with Split Horizon Poison Reverse



Router C learns about route X and advertises that route to Router B. Router B in turn advertises route X to Router A but sends a route X unreachable update back to Router C.

By default, split horizon is enabled on all interfaces.

Route Filtering

You can configure a route policy on a RIP-enabled interface to filter the RIP updates. Cisco NX-OS updates the route table with only those routes that the route policy allows.

Route Summarizations

You can configure multiple summary aggregate addresses for a specified interface. Route summarization simplifies route tables by replacing a number of more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

If more specific routes are in the routing table, RIP advertises the summary address from the interface with a metric equal to the maximum metric of the more specific routes.

Note Cisco NX-OS does not support automatic route summarization.

Route Redistributions

You can use RIP to redistribute static routes or routes from other protocols. You must configure a route map with the redistribution to control which routes are passed into RIP. A route policy allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. For more information, see Configuring Route Policy Manager.

Whenever you redistribute routes into a RIP routing domain, Cisco NX-OS does not, by default, redistribute the default route into the RIP routing domain. You can generate a default route into RIP, which can be controlled by a route policy.

You also configure the default metric that is used for all imported routes into RIP.

Load Balancing

You can use load balancing to allow a router to distribute traffic over all the router network ports that are the same distance from the destination address. Load balancing increases the usage of network segments and increases effective network bandwidth.

Cisco NX-OS supports the Equal Cost Multiple Paths (ECMP) feature with up to 16 equal-cost paths in the RIP route table and the unicast RIB. You can configure RIP to load balance traffic across some or all of those paths.

High Availability for RIP

Cisco NX-OS supports stateless restarts for RIP. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration, and RIP immediately sends request packets to repopulate its routing table.

Virtualization Support for RIP

Cisco NX-OS supports multiple instances of the RIP protocol that run on the same system. RIP supports virtual routing and forwarding (VRF) instances.

Prerequisites for RIP

RIP has the following prerequisites:

- You must enable RIP (see the Enabling RIP section).

Guidelines and Limitations for RIP

RIP Configuration Guidelines and Limitations

Follow these guidelines and limitations when configuring RIP.

- Names in the prefix-list are case-insensitive. We recommend using unique names. Do not use the same name by modifying upper-case and lower-case characters. For example, CTCPrimaryNetworks and CtcPrimaryNetworks are not two different entries.
- Cisco NX-OS does not support RIPv1. If Cisco NX-OS receives an RIPv1 packet, it logs a message and drops the packet.
- Cisco NX-OS does not establish adjacencies with RIPv1 routers.
- RIP is not supported in tunnel interfaces.
- IPv6 is not supported for RIP.

Note RIP only supports an 8-bit KeyID, that is less than or equal to 255. This is the keyID used while configuring authentication with RIP.

Default Settings for RIP Parameters

This section lists the default settings for RIP parameters.

Table 1: Default RIP Parameters

Parameters	Default
Maximum paths for load balancing	16
RIP feature	Disabled
Split horizon	Enabled

RIPs

If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enable RIP

You must enable RIP before you can configure RIP.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 (Optional) Use the [**no**] **feature rip** command to enable the RIP feature.

Example:

```
switch(config)# feature rip
```

Enables the RIP feature.

Step 3 (Optional) Use the **show feature** command to display enabled and disabled features.

Example:

```
switch(config)# show feature
```

Displays enabled and disabled features.

Step 4 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config
startup-config
```

Saves this configuration change.

Create a RIP Instance

Before you begin

You must enable RIP (see the Enabling RIP section).

You can create a RIP instance and configure the address family for that instance.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 (Optional) Use the **[no] router rip** *instance-tag* command to create a new RIP instance with the configured *instance-tag*.

Example:

```
switch(config)# router RIP Enterprise
switch(config-router)#
```

Step 3 (Optional) Use the **address-familyipv4unicast** command to configure the address family for this RIP instance and enter address-family configuration mode.

Example:

```
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)#
```

Step 4 (Optional) Use the **show ip rip** [**instance** *instance-tag*] [**vrf** *vrf-name*] command to display a summary of RIP information for all RIP instances.

Example:

```
switch(config-router-af)# show ip rip
```

Step 5 (Optional) Use the **distance** *value* command to set the administrative distance for RIP. The range is from 1 to 255. The default is 120. See the Administrative Distance section.

Example:

```
switch(config-router-af)# distance 30
```

Step 6 (Optional) Use the **maximum-paths** *number* command to configure the maximum number of equal-cost paths that RIP maintains in the route table. The range is from 1 to 64. The default is 16.

Example:

```
switch(config-router-af)# maximum-paths 6
```

Step 7 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-router-af)# copy running-config
startup-config
```

This example shows how to create a RIP instance for IPv4 and set the number of equal-cost paths for load balancing:

```
switch# configure terminal
switch(config)# router rip Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# max-paths 10
switch(config-router-af)# copy running-config startup-config
```

Restart the RIP Instance

You can restart a RIP instance and remove all associated neighbors for the instance.

To restart a RIP instance and remove all associated neighbors, use the following command in global configuration mode:

Procedure

(Optional) Use the **restart rip** *instance-tag* command.

Example:

```
switch(config)# restart rip Enterprise
```

Restarts the RIP instance and removes all neighbors.

Configure RIP on an Interface

Before you begin

You must enable RIP (see the Enabling RIP section).

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 (Optional) Use the **interface** *interface-type slot*/*port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/2
switch(config-if)#
```

Step 3 (Optional) Use the **ip router rip** *instance-tag* command to associate this interface with a RIP instance.

Example:

```
switch(config-if)# ip router rip
Enterprise
```

Step 4 (Optional) Use the **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*] [**detail**] command to display RIP information for an interface.

Example:

```
switch(config-if)# show ip rip
Enterprise tethernet 1/2
```

Step 5 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-if)# copy running-config
startup-config
```

This example shows how to add Ethernet 1/2 interface to a RIP instance:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip router rip Enterprise
switch(config)# copy running-config startup-config
```

Configure RIP Authentication

Before you begin

You must enable RIP (see the Enabling RIP section).

Configure a keychain if necessary before enabling authentication. For details about implementing keychains, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

You can configure authentication for RIP packets on an interface.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 (Optional) Use the **interface** *interface-type slot*/*port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/2
switch(config-if)#
```

Step 3 (Optional) Use the **ip rip authentication mode** {**text** | **md5**} command to set the authentication type for RIP on this interface as cleartext or MD5 authentication digest.

Example:

```
switch(config-if)# ip rip authentication
mode md5
```

Step 4 (Optional) Use the **ip rip authentication key-chain** *key* command to configure the authentication key used for RIP on this interface.

Example:

```
switch(config-if)# ip rip authentication key-chain RIPKey
```

Step 5 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-if)# copy running-config
startup-config
```

This example shows how to create a keychain and configure MD5 authentication on a RIP interface:

```
switch# configure terminal
switch(config)# key chain RIPKey
switch(config-keychain)# key 2
switch(config-keychain-key)# accept-lifetime 00:00:00 Jan 01 2000 infinite
switch(config-keychain-key)# send-lifetime 00:00:00 Jan 01 2000 infinite
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# interface ethernet 1/2
```

```
switch(config-if)# ip rip authentication mode md5
switch(config-if)# ip rip authentication key-chain RIPKey
switch(config-if)# copy running-config startup-config
```

Configure a Passive Interface

You can configure a RIP interface to receive routes but not send route updates by setting the interface to passive mode.

To configure a RIP interface in passive mode, use the following command in interface configuration mode:

Procedure

(Optional) Use the **ip rip passive-interface** command.

Example:

```
switch(config-if)# ip rip
passive-interface
```

Sets the interface to passive mode.

Configure Split Horizon With Poison Reverse

You can configure an interface to advertise routes learned by RIP as unreachable over the interface that learned the routes by enabling poison reverse.

To configure split horizon with poison reverse on an interface, use the following command in interface configuration mode:

Procedure

(Optional) Use the **ip rip poison-reverse** command.

Example:

```
switch(config-if)# ip rip poison-reverse
```

Enables split horizon with poison reverse. Split horizon with poison reverse is disabled by default.

Configure Route Summarization

You can create aggregate addresses that are represented in the routing table by a summary address. Cisco NX-OS advertises the summary address metric that is the smallest metric of all the more specific routes.

To configure a summary address on an interface, use the following command in interface configuration mode:

Procedure

(Optional) Use the **iprip summary-address***ip-prefix/mask-len* command.

Example:

```
switch(config-if)# ip rip summary-address 1.1.1.1/32
```

Configures a summary address for RIP for IPv4 addresses.

Configure Route Redistribution

Before you begin

You must enable RIP (see the Enabling RIP section)

Configure a route map before configuring redistribution . See the Configuring Route Maps section for details on configuring route maps.

You can configure RIP to accept routing information from another routing protocol and redistribute that information through the RIP network. Redistributed routes can be assigned a default route.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 (Optional) Use the **router rip** *instance-tag* command to create a new RIP instance with the configured *instance-tag*.

Example:

```
switch(config)# router rip Enterprise
switch(config-router)#
```

Step 3 (Optional) Use the **address-family ipv4 unicast** command to enter address-family configuration mode.

Example:

```
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)#
```

Step 4 (Optional) Use the **redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name* command to redistribute routes from other protocols into RIP.

Example:

```
switch(config-router-af)# redistribute
eigrp 201 route-map RIPmap
```

Step 5 (Optional) Use the **default-information originate** [**always**] [**route-map** *map-name*] command to generate a default route into RIP, controlled by a route map if required.

Example:

```
switch(config-router-af)#
default-information originate always
```

Step 6 (Optional) Use the **default-metric** *value* command to set the default metric for all redistributed routes.

Example:

```
switch(config-router-af)# default-metric
2
```

Step 7 (Optional) Use the **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**]] [**vrf** *vrf-name*] [**summary**] command to show the routes in RIP.

Example:

```
switch(config-router-af)# show ip rip
route
```

Step 8 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-router-af)# copy running-config
startup-config
```

This example shows how to redistribute EIGRP into RIP:

```
switch# configure terminal
switch(config)# router rip Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map RIPmap
switch(config-router-af)# copy running-config startup-config
```

Configure Cisco NX-OS RIP for Compatibility with Cisco IOS RIP

Before you begin

You must enable RIP (see the Enabling RIP section).

You can configure Cisco NX-OS RIP to behave like Cisco IOS RIP in the way that routes are advertised and processed.

Directly connected routes are treated with cost 1 in Cisco NX-OS RIP and with cost 0 in Cisco IOS RIP. When routes are advertised in Cisco NX-OS RIP, the receiving device adds a minimum cost of +1 to all received routes and installs the routes in its routing table. In Cisco IOS RIP, this cost increment is done on the sending router, and the receiving router installs the routes without any modification. This difference in behavior can cause issues when both Cisco NX-OS and Cisco IOS devices are working together. You can prevent these compatibility issues by configuring Cisco NX-OS RIP to advertise and process routes like Cisco IOS RIP.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Configure the RIP instance and compatibility settings.

a) Use the **router rip** *instance-tag* command to create a new RIP instance with the configured instance tag.

Example:

```
switch(config)# router rip 100
switch(config-router)#
```

Creates a new RIP instance with the configured instance tag. You can enter 100, 201, or up to 20 alphanumeric chapters for the instance tag.

b) Use the [**no**] **metric direct 0** command to configure all directly connected routes with cost 0 for Cisco IOS RIP compatibility.

Example:

```
switch(config-router)# metric direct 0
```

Configures all directly connected routes with cost 0 instead of the default of cost 1 in order to make Cisco NX-OS RIP compatible with Cisco IOS RIP in the way that routes are advertised and processed.

Note
This command must be configured on all Cisco NX-OS devices that are present in any RIP network that also contains Cisco IOS devices.

c) Use the **address-family ipv4 unicast** command to configure the VRF address family for this RIP instance.

Example:

```
switch(config-router)# address-family ipv4 unicast
```

Configures the VRF address family for this RIP instance.

d) Use the **vrf abc** command to create a new VRF.

Example:

```
switch(config-router)# vrf abc
```

Creates a new VRF.

e) Use the **flush-routes** command to remove routes.

Example:

```
switch(config-router)# flush-routes
```

Remove routes.

f) Use the **isolate** command to isolate the router from RIP perspective.

Example:

```
switch(config-router)# isolate
```

Isolates router from RIP perspective.

g) Use the **shutdown** command to shut down the RIP instance.

Example:

```
switch(config-router)# shutdown
```

Shuts down the RIP instance.

Note
The show run output will differ if the shutdown command configured is under process and a vrf is also created. The shutdown configuration will be shown above vrf configuration.

Step 3 (Optional) Use the **show running-config rip** command to display the current running RIP configuration.

Example:

```
switch(config-router)# show
running-config rip
!Command: show running-config rip
!Running configuration last done at: Tue Dec 17 01:13:28 2024
!Time: Tue Dec 17 01:13:31 2024
```

Displays the current running RIP configuration.

Step 4 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-router)# copy running-config
startup-config
```

Saves this configuration change.

This example shows how to disable Cisco NX-OS RIP compatibility with Cisco IOS RIP by returning all direct routes from cost 0 to cost 1:

```
switch# configure terminal
switch(config)# router rip 100
switch(config-router)# no metric direct 0
switch(config-router)# show running-config rip
switch(config-router)# copy running-config startup-config
```

Configure Virtualization

Before you begin

You must enable RIP (see the Enabling RIP section).kaanipakaa

You can configure multiple RIP instances, create multiple VRFs, and use the same or multiple RIP instances in each VRF. You assign a RIP interface to a VRF.

Note Configure all other parameters for an interface after you configure the VRF for an interface. Configuring a VRF for an interface deletes all the configurations for that interface.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Configure the VRF context and RIP instance.

a) (Optional) Use the **vrf context** *vrf-name* command to create a new VRF and enter VRF configuration mode.

Example:

```
switch(config)# vrf context RemoteOfficeVRF
switch(config-vrf)#
```

b) (Optional) Use the **exit** command to exit VRF configuration mode.

Example:

```
switch(config-vrf)# exit
switch(config)#
```

c) (Optional) Use the **router rip** *instance-tag* command to create a new RIP instance with the configured instance tag.

Example:

```
switch(config)# router rip Enterprise
switch(config-router)#
```

d) (Optional) Use the **vrf** *vrf-name* command to create a new VRF.

Example:

```
switch(config-router)# vrf RemoteOfficeVRF
switch(config-router-vrf)#
```

e) (Optional) Use the **address-familyipv4unicast** command to configure the VRF address family for this RIP instance.

Example:

```
switch(config-router-vrf)# address-family ipv4 unicast
switch(config-router-vrf-af)#
```

f) (Optional) Use the **redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name* command to redistribute routes from other protocols into RIP.

Example:

```
switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap
```

See Configure route maps for more information about route maps.

Step 3 Configure the interface and assign it to the VRF.

a) (Optional) Use the **interface ethernet** *slot*/*port* command to enter interface configuration mode.

Example:

```
switch(config-router-vrf-af)# interface ethernet 1/2
switch(config-if)#
```

b) (Optional) Use the **vrf member** *vrf-name* command to add this interface to a VRF.

Example:

```
switch(config-if)# vrf member RemoteOfficeVRF
```

c) (Optional) Use the **ip address** *ip-prefix*/*length* command to configure an IP address for this interface.

Example:

```
switch(config-if)# ip address 192.0.2.1/16
```

You must perform this step after you assign this interface to a VRF.

d) (Optional) Use the **ip router rip** *instance-tag* command to associate this interface with a RIP instance.

Example:

```
switch(config-if)# ip router rip Enterprise
```

Step 4 (Optional) Use the **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*] command to display RIP information for an interface in a VRF.

Example:

```
switch(config-if)# show ip rip Enterprise ethernet 1/2
```

Step 5 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-if)# copy running-config startup-config
```

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal
switch(config)# vrf context RemoteOfficeVRF
switch(config-vrf)# exit
switch(config)# router rip Enterprise
switch(config-router)# vrf RemoteOfficeVRF
switch(config-router-vrf)# address-family ipv4 unicast
switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap
switch(config-router-vrf-af)# interface ethernet 1/2
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router rip Enterprise
switch(config-if)# copy running-config startup-config
```

Tune the RIP Protocol

You can tune RIP to match your network requirements. RIP uses several timers that determine the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internetwork needs.

Note You must configure the same values for the RIP timers on all RIP-enabled routers in your network.

Procedure

Step 1 (Optional) Use the **timers basic** *update timeout holddown garbage-collection* command to set the RIP timers in seconds.

Example:

```
switch(config-router-af)# timers basic 40 120 120 100
```

The parameters are as follows:

- *update*—The range is from 5 to any positive integer. The default is 30.
- *timeout*—The time that Cisco NX-OS waits before declaring a route as invalid. If Cisco NX-OS does not receive route update information for this route before the timeout interval ends, Cisco NX-OS declares the route as invalid. The range is from 1 to any positive integer. The default is 180.
- *holddown*—The time during which Cisco NX-OS ignores better route information for an invalid route. The range is from 0 to any positive integer. The default is 180.
- *garbage-collection*—The time from when Cisco NX-OS marks a route as invalid until Cisco NX-OS removes the route from the routing table. The range is from 1 to any positive integer. The default is 120.

Step 2 (Optional) Use the **ip rip metric-offset** *value* command to add a value to the metric for every route received on this interface.

Example:

```
switch(config-if)# ip rip metric-offset 10
```

The range is from 1 to 15. The default is 1.

Step 3 (Optional) Use the **ip rip route-filter** {**prefix-list** *list-name* | **route-map** *map-name* | [**in** | **out**]} command to specify a route map to filter incoming or outgoing RIP updates.

Example:

```
switch(config-if)# ip rip route-filter route-map
            InputMap in
```

Verify the RIP Configuration

Procedure

Step 1 Use the **show ip rip instance** [*instance-tag*] [**vrf** *vrf-name*] command to display the status for an instance of RIP.

Example:

```
show ip rip instance [instance-tag] [vrf vrf-name]
```

Step 2 Use the **show ip rip** [**instance** *instance-tag*] **interface** *slot/port* **detail** [**vrf** *vrf-name*] command to display the RIP status for an interface.

Example:

```
show ip rip [instance instance-tag] interface slot/port detail [vrf vrf-name]
```

Step 3 Use the **show ip rip** [**instance** *instance-tag*] **neighbor** [*interface-type number*] [**vrf** *vrf-name*] command to display the RIP neighbor table.

Example:

```
show ip rip [instance instance-tag] neighbor [interface-type number] [vrf vrf-name]
```

Step 4 Use the **show ip rip** [**instance** *instance-tag*] **route** [*ip-prefix/length* [**longer-prefixes** | **shorter-prefixes**]] [**summary**] [**vrf** *vrf-name*] command to display the RIP route table.

Example:

```
show ip rip [instance instance-tag] route [ip-prefix/length [longer-prefixes |
shorter-prefixes]] [summary] [vrf vrf-name]
```

Step 5 Use the **show running-configuration rip** command to display the current running RIP configuration.

Example:

```
show running-configuration rip
```

Display the RIP Statistics

Procedure

Step 1 Use the **show ip rip** [**instance** *instance-tag*] **policy statistics redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} [**vrf** *vrf-name*] command to display the RIP policy statistics.

Example:

```
switch# show ip rip [instance instance-tag] policy statistics redistribute {bgp as | direct
 | {eigrp | isis | ospf | ospfv3 | rip} instance-tag | static} [vrf vrf-name]
RIP Policy Statistics Output...
```

Step 2 Use the **show ip rip** [**instance** *instance-tag*] **statistics** *interface-type number* [**vrf** *vrf-name*] command to display the RIP statistics.

Example:

```
switch# show ip rip [instance instance-tag] statistics interface-type number [vrf vrf-name]
RIP Statistics Output...
```

Step 3 Use the **clear rip policy statistics redistribute** *protocol process-tag* command to clear policy statistics.

Example:

```
switch# clear rip policy statistics redistribute protocol process-tag
Policy statistics cleared.
```

Step 4 Use the **clear ip rip statistics** command to clear RIP statistics.

Example:

```
switch# clear ip rip statistics
RIP statistics cleared.
```

Configuration Examples for RIP

Configuration Examples for RIP.

Configuration Examples

The following example shows how to create the Enterprise RIP instance in a VRF and add Ethernet interface 1/2 to this RIP instance. The example also shows how to configure authentication for Ethernet interface 1/2 and redistribute EIGRP into this RIP domain.

```
vrf context NewVRF
!
 feature rip
 router rip Enterprise
  vrf NewVRF
   address-family ipv4 unicast
    redistribute eigrp 201 route-map RIPmap
    maximum-paths 10
!
interface ethernet 1/2
 vrf member NewVRF
 ip address 192.0.2.1/16
 ip router rip Enterprise
 ip rip authentication mode md5
 ip rip authentication key-chain RIPKey
```

The following example shows a valid keyID configuration:

```
### Valid
key-chain kc1
key 255
key-string ...
```

Related Topics

See Configuring Route Policy Manager for more information on route maps.