Implementing RIP for IPv6

First Published: June 7, 2001
Last Updated: November 20, 2009

This module describes how to configure Routing Information Protocol for IPv6. RIP is a distance-vector routing protocol that uses hop count as a routing metric. RIP is an Interior Gateway Protocol (IGP) most commonly used in smaller networks.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 for Implementing RIP for IPv6" section on page 16.

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

Contents

Prerequisites for Implementing RIP for IPv6

- This module assumes that you are familiar with IPv6 addressing and basic configuration. Refer to *Implementing IPv6 Addressing and Basic Connectivity* for more information.
- This module assumes that you are familiar with IPv4. Refer to the publications referenced in the "Related Documents" section for IPv4 configuration and command reference information, as needed.

Information About Implementing RIP for IPv6

To configure IPv6 RIP, you need to understand the following concepts:

- RIP for IPv6, page 2
- Nonstop Forwarding for IPv6 RIP, page 2

RIP for IPv6

IPv6 RIP functions the same and offers the same benefits as RIP in IPv4. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes, and the use of the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages. New commands specific to RIP in IPv6 were also added to the Cisco IOS command-line interface (CLI).

In the Cisco IOS software implementation of IPv6 RIP each IPv6 RIP process maintains a local routing table, referred to as a Routing Information Database (RIB). The IPv6 RIP RIB contains a set of best-cost IPv6 RIP routes learned from all its neighboring networking devices. If IPv6 RIP learns the same route from two different neighbors, but with different costs, it will store only the lowest cost route in the local RIB. The RIB also stores any expired routes that the RIP process is advertising to its neighbors running RIP. IPv6 RIP will try to insert every non-expired route from its local RIB into the master IPv6 RIB. If the same route has been learned from a different routing protocol with a better administrative distance than IPv6 RIP, the RIP route will not be added to the IPv6 RIB but the RIP route will still exist in the IPv6 RIP RIB.

Nonstop Forwarding for IPv6 RIP

Cisco nonstop forwarding (NSF) continues forwarding packets while routing protocols converge, therefore avoiding a route flap on switchover. When an RP failover occurs, the Forwarding Information Base (FIB) marks installed paths as stale by setting a new epoch. Subsequently, the routing protocols reconverge and populate the RIB and FIB. Once all NSF routing protocols converge, any stale routes held in the FIB are removed. A failsafe timer is required to delete stale routes, in case of routing protocol failure to repopulate the RIB and FIB.

RIP registers as an IPv6 NSF client. Doing so has the benefit of using RIP routes installed in the Cisco Express Forwarding table until RIP has converged on the standby.

How to Implement RIP for IPv6

When configuring supported routing protocols in IPv6, you must create the routing process, enable the routing process on interfaces, and customize the routing protocol for your particular network.

Note The following sections describe the configuration tasks for creating an IPv6 RIP routing process and enabling the routing process on interfaces. The following sections do not provide in-depth information on customizing RIP because the protocol functions the same in IPv6 as it does in IPv4. Refer to the publications referenced in the "Related Documents" section for further IPv6 and IPv4 configuration and command reference information.

The tasks in the following sections explain how to configure IPv6 RIP. Each task in the list is identified as either required or optional:

This section contains the following procedures:

Enabling the IPv6 RIP Process

This task explains how to enable the specified IPv6 RIP process on an interface.

Prerequisites

Before configuring the router to run IPv6 RIP, globally enable IPv6 using the **ipv6 unicast-routing** command in global configuration mode, and enable IPv6 on any interfaces on which IPv6 RIP is to be enabled. For details on basic IPv6 connectivity tasks, refer to the *Implementing Basic Connectivity for IPv6* module.

If you want to set or change a global value, follow steps 1 and 2, and then use the optional **ipv6 router rip** command in global configuration mode (see Customizing IPv6 RIP, page 4 for an example).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 rip** *name* **enable**

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	**ipv6 unicast-routing** **Example:** `Router(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 5	**ipv6 rip** *name* **enable** **Example:** `Router(config-if)# ipv6 rip process1 enable`	Enables the specified IPv6 RIP routing process on an interface.

Customizing IPv6 RIP

This optional task explains how to customize IPv6 RIP by configuring the maximum numbers of equal-cost paths that IPv6 RIP will support, adjusting the IPv6 RIP timers, and originating a default IPv6 route.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router rip** *word*
4. **maximum-paths** *number-paths*
5. **exit**
6. **interface** *type number*
7. **ipv6 rip** *name* **default-information** {**only** | **originate**} [**metric** *metric-value*]

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	**ipv6 router rip** *word* **Example:** `Router(config)# ipv6 router rip process1`	Configures an IPv6 RIP routing process and enters router configuration mode for the IPv6 RIP routing process. • Use the *word* argument to identify a specific IPv6 RIP routing process.
Step 4	**maximum-paths** *number-paths* **Example:** `Router(config-router)# maximum-paths 1`	(Optional) Defines the maximum number of equal-cost routes that IPv6 RIP can support. • The *number-paths* argument is an integer from 1 to 64. The default for RIP is four paths.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 7	**ipv6 rip** *name* **default-information** {**only** | **originate**} [**metric** *metric-value*] **Example:** `Router(config-if)# ipv6 rip process1 default-information originate`	(Optional) Originates the IPv6 default route (::/0) into the specified RIP routing process updates sent out of the specified interface. **Note** To avoid routing loops after the IPv6 default route (::/0) is originated out of any interface, the routing process ignores all default routes received on any interface. • Specifying the **only** keyword originates the default route (::/0) but suppresses all other routes in the updates sent on this interface. • Specifying the **originate** keyword originates the default route (::/0) in addition to all other routes in the updates sent on this interface.

Redistributing Routes into an IPv6 RIP Routing Process

RIP supports the use of a route map to select routes for redistribution. Routes may be specified by prefix, using a route-map prefix list, or by tag, using the route-map “match tag” function.

The maximum metric that RIP can advertise is 16, and a metric of 16 denotes a route that is unreachable. Therefore, if you are redistributing routes with metrics greater than or equal to 16, then by default RIP will advertise them as unreachable. These routes will not be used by neighboring routers. The user must configure a redistribution metric of less than 15 for these routes.

Note You must to advertise a route with metric of 15 or less. A RIP router always adds an interface cost—the default is 1—onto the metric of a received route. If you advertise a route with metric 15, your neighbor will add 1 to it, making a metric of 16. Because a metric of 16 is unreachable, your neighbor will not install the route in the routing table.

If no metric is specified, then the current metric of the route is used. To find the current metric of the route, enter the **show ipv6 route** command.

This task explains how to redistribute routes into an IPv6 RIP routing process.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 rip** *name* **enable**
5. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** {**internal** | **external**}] [**route-map** *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	`interface` *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies the interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ipv6 rip** *word* **enable** **Example:** `Router(config-if)# ipv6 router one enable`	Enables an IPv6 Routing Information Protocol (RIP) routing process on an interface.
Step 5	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** {**internal** | **external**}] [**route-map** *map-name*] **Example:** `Router(config-router)# redistribute bgp 65001 route-map bgp-to-rip`	Redistributes the specified routes into the IPv6 RIP routing process. • The *protocol* argument can be one of the following keywords: **bgp**, **connected**, **isis**, **rip**, or **static**. • The **rip** keyword and *process-id* argument specify an IPv6 RIP routing process. **Note** The **connected** keyword refers to routes that are established automatically by assigning IPv6 addresses to an interface.

Configuring Route Tags for IPv6 RIP Routes

When performing route redistribution, you can associate a numeric tag with a route. The tag is advertised with the route by RIP and will be installed along with the route in neighboring router's routing table.

If you redistribute a tagged route (for example, a route in the IPv6 routing table that already has a tag) into RIP, then RIP will automatically advertise the tag with the route. If you use a redistribution route map to specify a tag, then RIP will use the route map tag in preference to the routing table tag.

The following task explains how to set route tags using a route map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}
5. **set tag** tag-*value*

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.

	Command or Action	Purpose
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map bgp-to-rip permit 10`	Defines a route map, and enters route-map configuration mode. • Follow this step with a **match** command.
Step 4	**match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*} **Example:** `Router(config-route-map)# match ipv6 address prefix-list bgp-to-rip-flt`	Specifies a list of IPv6 prefixes to be matched.
Step 5	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 4`	Sets the tag value to associate with the redistributed routes.

Filtering IPv6 RIP Routing Updates

Route filtering using distribute lists provides control over the routes RIP receives and advertises. This control may be exercised globally or per interface.

This task explains how to filter IPv6 RIP routing updates by applying a prefix list to IPv6 RIP routing updates that are received or sent on an interface.

IPv6 Distribute Lists

Filtering is controlled by distribute lists. Input distribute lists control route reception, and input filtering is applied to advertisements received from neighbors. Only those routes that pass input filtering will be inserted in the RIP local routing table and become candidates for insertion into the IPv6 routing table.

Output distribute lists control route advertisement; Output filtering is applied to route advertisements sent to neighbors. Only those routes passing output filtering will be advertised.

Global distribute lists (which are distribute lists that do not apply to a specified interface) apply to all interfaces. If a distribute list specifies an interface, then that distribute list applies only to that interface.

An interface distribute list always takes precedence. For example, for a route received at an interface, with the interface filter set to deny, and the global filter set to permit, the route is blocked, the interface filter is passed, the global filter is blocked, and the route is passed.

IPv6 Prefix List Operand Keywords

IPv6 prefix lists are used to specify certain prefixes or a range of prefixes that must be matched before a permit or deny statement can be applied. Two operand keywords can be used to designate a range of prefix lengths to be matched. A prefix length of less than, or equal to, a value is configured with the **le** keyword. A prefix length greater than, or equal to, a value is specified using the **ge** keyword. The **ge** and **le** keywords can be used to specify the range of the prefix length to be matched in more detail than the usual *ipv6-prefix*/*prefix-length* argument. For a candidate prefix to match against a prefix list entry three conditions can exist:

- The candidate prefix must match the specified prefix list and prefix length entry.

- The value of the optional **le** keyword specifies the range of allowed prefix lengths from the *prefix-length* argument up to, and including, the value of the **le** keyword.
- The value of the optional **ge** keyword specifies the range of allowed prefix lengths from the value of the **ge** keyword up to, and including, 128.

Note Note that the first condition must match before the other conditions take effect.

An exact match is assumed when the **ge** or **le** keywords are not specified. If only one keyword operand is specified then the condition for that keyword is applied, and the other condition is not applied. The *prefix-length* value must be less than the **ge** value. The **ge** value must be less than, or equal to, the **le** value. The **le** value must be less than or equal to 128.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 prefix list** *prefix-list-name* [**seq** *seq-number*] {**deny** *ipv6-prefix***/***prefix-length* **|** **description** *text*} [**ge** *ge-value*] [**le** *le-value*]
4. **ipv6 prefix list** *prefix-list-name* [**seq** *seq-number*] {**permit** *ipv6-prefix***/***prefix-length* **|** **description** *text*} [**ge** *ge-value*] [**le** *le-value*]
5. Repeat Steps 3 and 4 as many times as necessary to build the prefix list.
6. **ipv6 router rip** *name*
7. **distribute-list prefix-list** *prefix-list-name* {**in** **|** **out**} [*interface-type interface-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	**ipv6 prefix list** *prefix-list-name* [**seq** *seq-number*] {**deny** *ipv6-prefix***/***prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ipv6 prefix-list abc permit 2001:0db8::/16`	Creates an entry in the IPv6 prefix list.

	Command or Action	Purpose
Step 4	**ipv6 prefix list** *prefix-list-name* [**seq** *seq-number*] {**deny** *ipv6-prefix***/***prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ipv6 prefix-list abc deny ::/0`	Creates an entry in the IPv6 prefix list.
Step 5	Repeat Steps 3 and 4 as many times as necessary to build the prefix list.	—
Step 6	**ipv6 router rip** *name* **Example:** `Router(config)# ipv6 router rip process1`	Configures an IPv6 RIP routing process.
Step 7	**distribute-list prefix-list** *prefix-list-name* {**in** | **out**} [*interface-type interface-number*] **Example:** `Router(config-rtr-rip)# distribute-list prefix-list process1 in ethernet 0/0`	Applies a prefix list to IPv6 RIP routing updates that are received or sent on an interface.

Verifying IPv6 RIP Configuration and Operation

A user may want to check IPv6 RIP configuration and operation. This task explains how to display information to verify the configuration and operation of IPv6 RIP.

SUMMARY STEPS

1. **show ipv6 rip** [*name*] [**database** | **next-hops**]
2. **show ipv6 route** [*ipv6-address* | *ipv6-prefix***/***prefix-length* | *protocol* | *interface-type interface-number*]
3. **enable**
4. **debug ipv6 rip** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ipv6 rip** [*name*] [**database** | **next-hops**] **Example:** `Router> show ipv6 rip process1 database`	(Optional) Displays information about current IPv6 RIP processes. • In this example, IPv6 RIP process database information is displayed for the specified IPv6 RIP process.
Step 2	**show ipv6 route** [*ipv6-address* | *ipv6-prefix***/***prefix-length* | *protocol* | *interface-type interface-number*] **Example:** `Router> show ipv6 route rip`	(Optional) Displays the current contents of the IPv6 routing table. • In this example, only IPv6 RIP routes are displayed.

	Command or Action	Purpose
Step 3	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 4	**debug ipv6 rip** [*interface-type interface-number*] **Example:** `Router# debug ipv6 rip`	(Optional) Displays debugging messages for IPv6 RIP routing transactions.

Examples

This section provides the following output examples:

- Sample Output from the show ipv6 rip Command, page 11
- Sample Output from the show ipv6 route Command, page 12
- Sample Output from the debug ipv6 rip Command, page 12

Sample Output from the show ipv6 rip Command

In the following example, output information about all current IPv6 RIP processes is displayed using the **show ipv6 rip** command:

```
Router> show ipv6 rip

RIP process "process1", port 521, multicast-group FF02::9, pid 62
     Administrative distance is 120. Maximum paths is 1
     Updates every 5 seconds, expire after 15
     Holddown lasts 10 seconds, garbage collect after 30
     Split horizon is on; poison reverse is off
     Default routes are generated
     Periodic updates 223, trigger updates 1
  Interfaces:
    Ethernet0/0
  Redistribution:
    Redistributing protocol bgp 65001 route-map bgp-to-rip
```

In the following example, output information about a specified IPv6 RIP process database is displayed using the **show ipv6 rip** command with the *name* argument and the **database** keyword. In the following output for the IPv6 RIP process named process1, timer information is displayed, and route 2001:0db8::16/64 has a route tag set:

```
Router> show ipv6 rip process1 database

RIP process "process1", local RIB
 2001:0db8::/64, metric 2
     Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:0db8::/16, metric 2 tag 4, installed
     Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:0db8:1::/16, metric 2 tag 4, installed
     Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 2001:0db8:2::/16, metric 2 tag 4, installed
     Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
 ::/0, metric 2, installed
     Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
```

In the following example, output information for a specified IPv6 RIP process is displayed using the **show ipv6 rip** user EXEC command with the *name* argument and the **next-hops** keyword:

```
Router> show ipv6 rip process1 next-hops

RIP process "process1", Next Hops
  FE80::A8BB:CCFF:FE00:A00/Ethernet0/0 [4 paths]
```

Sample Output from the show ipv6 route Command

The current metric of the route can be found by entering the **show ipv6 route** command. In the following example, output information for all IPv6 RIP routes is displayed using the **show ipv6 route** command with the **rip** protocol keyword:

```
Router> show ipv6 route rip

IPv6 Routing Table - 17 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
R   2001:0db8:1::/32 [120/2]
     via FE80::A8BB:CCFF:FE00:A00, Ethernet0/0
R   2001:0db8:2::/32 [120/2]
     via FE80::A8BB:CCFF:FE00:A00, Ethernet0/0
R   2001:0db8:3::/32 [120/2]
     via FE80::A8BB:CCFF:FE00:A00, Ethernet0/0
```

Sample Output from the debug ipv6 rip Command

In the following example, debugging messages for IPv6 RIP routing transactions are displayed using the **debug ipv6 rip** command:

Note By default, the system sends the output from **debug** commands and system error messages to the console. To redirect debugging output, use the **logging** command options within privileged EXEC mode. Possible destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server.

```
Router# debug ipv6 rip

RIPng: Sending multicast update on Ethernet0/0 for process1
       src=FE80::A8BB:CCFF:FE00:B00
       dst=FF02::9 (Ethernet0/0)
       sport=521, dport=521, length=112
       command=2, version=1, mbz=0, #rte=5
       tag=0, metric=1, prefix=2001:0db8::/64
       tag=4, metric=1, prefix=2001:0db8:1::/16
       tag=4, metric=1, prefix=2001:0db8:2;:/16
       tag=4, metric=1, prefix=2001:0db8:3::/16
       tag=0, metric=1, prefix=::/0
RIPng: Next RIB walk in 10032
RIPng: response received from FE80::A8BB:CCFF:FE00:A00 on Ethernet0/0 for process1
       src=FE80::A8BB:CCFF:FE00:A00 (Ethernet0/0)
       dst=FF02::9
       sport=521, dport=521, length=92
       command=2, version=1, mbz=0, #rte=4
       tag=0, metric=1, prefix=2001:0db8::/64
       tag=0, metric=1, prefix=2001:0db8:1::/32
       tag=0, metric=1, prefix=2001:0db8:2::/32
       tag=0, metric=1, prefix=2001:0db8:3::/32
```

Configuration Examples for IPv6 RIP

This section provides the following configuration example:

- IPv6 RIP Configuration: Example, page 13

IPv6 RIP Configuration: Example

In the following example, the IPv6 RIP process named process1 is enabled on the router and on Ethernet interface 0/0. The IPv6 default route (::/0) is advertised in addition to all other routes in router updates sent on Ethernet interface 0/0. Additionally, BGP routes are redistributed into the RIP process named process1 according to a route map where routes that match a prefix list are also tagged. The number of parallel paths is set to one to allow the route tagging, and the IPv6 RIP timers are adjusted. A prefix list named eth0/0-in-flt filters inbound routing updates on Ethernet interface 0/0.

```
ipv6 router rip process1
 maximum-paths 1
 redistribute bgp 65001 route-map bgp-to-rip
 distribute-list prefix-list eth0/0-in-flt in Ethernet0/0
!
interface Ethernet0/0
 ipv6 address 2001:0db8::/64 eui-64
 ipv6 rip process1 enable
 ipv6 rip process1 default-information originate
!
ipv6 prefix-list bgp-to-rip-flt seq 10 deny 2001:0db8:3::/16 le 128
ipv6 prefix-list bgp-to-rip-flt seq 20 permit 2001:0db8:1::/8 le 128
!
ipv6 prefix-list eth0/0-in-flt seq 10 deny ::/0
ipv6 prefix-list eth0/0-in-flt seq 15 permit ::/0 le 128
!
route-map bgp-to-rip permit 10
 match ipv6 address prefix-list bgp-to-rip-flt
 set tag 4
```

Where to Go Next

If you want to implement more IPv6 routing protocols, see the *Implementing IS-IS for IPv6* or *Implementing Multiprotocol BGP for IPv6* module.

Additional References

The following sections provide references related to the Implementing RIP for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv4 RIP configuration tasks	"Configuring Routing Information Protocol," *Cisco IOS IP Routing Protocols Configuration Guide*
RIP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	"RIP Commands," *Cisco IOS IP Routing Protocols Command Reference*
IPv6 supported feature list	"Start Here: Cisco IOS Software Release Specifics for IPv6 Features," *Cisco IOS IPv6 Configuration Guide*
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IPv6 Command Reference*

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
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	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

RFCs	Title
RFC 2080	*RIPng for IPv6*

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 Implementing RIP for IPv6

Table 10 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the *Start Here: Cisco IOS Software Release Specifies for IPv6 Features* roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 10 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 10 Feature Information for Implementing RIP for IPv6

Feature Name	Releases	Feature Information
IPv6 routing: RIP for IPv6 (RIPng)	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	RIP enhancements for IPv6 include support for IPv6 addresses and prefixes, and the use of the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages. This entire document provides information about this features.
IPv6 routing: route redistribution	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	Routes may be specified by prefix, using a route-map prefix list, or by tag, using the route-map “match tag” function. The following sections provide information about this feature: • Redistributing Routes into an IPv6 RIP Routing Process, page 6 • Configuring Route Tags for IPv6 RIP Routes, page 7 • IPv6 RIP Configuration: Example, page 13
IPv6: RIPng Nonstop Forwarding	12.2(33)SRE	The IPv6 RIPng nonstop forwarding feature is supported. The following section provides information about this feature: • Nonstop Forwarding for IPv6 RIP, page 2