



IPv6 for Cisco IOS Software, File 2 of 3: Configuring

This document provides configuration tasks for the Cisco implementation of IP version 6 (IPv6) in the Cisco IOS software and includes the following sections:

- [Documentation Specifics, page 1](#)
- [Enabling IPv6 Routing and Configuring IPv6 Addressing, page 2](#)
- [Configuring IPv6 ICMP Rate Limiting, page 6](#)
- [Configuring Static Cache Entries for IPv6 Neighbor Discovery, page 7](#)
- [Configuring IPv6 Duplicate Address Detection, page 9](#)
- [Configuring IPv6 Redirect Messages, page 11](#)
- [Mapping Host Names to IPv6 Addresses, page 12](#)
- [Configuring IPv6 ATM and Frame Relay Interfaces, page 15](#)
- [Configuring IPv6 Standard Access Lists, page 21](#)
- [Configuring IPv6 RIP, page 24](#)
- [Configuring IPv6 IS-IS, page 28](#)
- [Configuring Multiprotocol BGP Extensions for IPv6, page 39](#)
- [Configuring Dual Protocol Stacks and IPv6 Overlay Tunnels, page 51](#)
- [Monitoring and Maintaining IPv6, page 57](#)

Documentation Specifics

IPv6 features are supported only in the 12.0 ST and 12.2 T Cisco IOS software release trains, starting at Cisco IOS Release 12.0(21)ST and Release 12.2(2)T, respectively. Subsequent releases of the 12.0 ST and 12.2 T Cisco IOS software trains will support additional IPv6 features. This document along with the following IPv6 for Cisco IOS Software feature documentation in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com will be updated with information about additional IPv6 features at each subsequent release of the applicable Cisco IOS software release trains:

- *Start Here: Cisco IOS Software Release Specifics for IPv6 Features*
- *IPv6 for Cisco IOS Software, File 1 of 3: Overview*
- *IPv6 for Cisco IOS Software, File 2 of 3: Configuring (This document)*
- *IPv6 for Cisco IOS Software, File 3 of 3: Commands*

**Note**

The *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document details which IPv6 features are supported in each release of the 12.0 ST and 12.2 T Cisco IOS software trains. *Not all IPv6 features may be supported in your Cisco IOS software release.* We strongly recommend that you read the entire *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document before reading the other IPv6 for Cisco IOS Software feature documentation.

The other IPv6 for Cisco IOS Software feature documentation provides IPv6 overview, configuration, and command reference information for IPv6 features in each release of the 12.0 ST and 12.2 T Cisco IOS software trains.

Enabling IPv6 Routing and Configuring IPv6 Addressing

By default, IPv6 routing is disabled in the Cisco IOS software. To enable IPv6 routing, you must first enable the forwarding of IPv6 traffic globally on the router and then you must assign IPv6 addresses to individual interfaces in the router.

The tasks described in the following sections explain how to enable IPv6 routing on a Cisco router. Each task in the list is identified as either required or optional:

- [Enabling IPv6 Processing Globally on the Router](#) (required)
- [Configuring IPv6 Addresses](#) (required)
- [Verifying IPv6 Operation and Address Configuration](#) (optional)

See the “[IPv6 Routing and IPv6 Address Configuration Example](#)” section for a configuration example.

Enabling IPv6 Processing Globally on the Router

To enable the forwarding of IPv6 traffic globally on the router, use the following command in global configuration mode:

Command	Purpose
Router(config)# <code>ipv6 unicast-routing</code>	Enables the forwarding of IPv6 unicast datagrams.

Configuring IPv6 Addresses

An IPv6 address must be configured on an interface for the interface to forward IPv6 traffic. Configuring a site-local or global IPv6 address on an interface automatically configures a link-local address and activates IPv6 for that interface. Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast and anycast address assigned to the interface
- All-nodes multicast group FF02:0:0:0:0:0:0:1 (scope is link-local)
- All-routers multicast group FF02:0:0:0:0:0:0:2 (scope is link-local)



Note The solicited-node multicast address is used in the neighbor discovery process.

To configure an IPv6 address on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface <i>interface-type</i> <i>interface-number</i>	Specifies an interface type and number, and places the router in interface configuration mode.
Step 2	Router(config-if)# ipv6 address <i>ipv6-prefix/prefix-length</i> [eui-64]	<p>Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface.</p> <p>Specifying the ipv6 address <i>ipv6-prefix/prefix-length</i> interface configuration command without the eui-64 keyword configures site-local and global IPv6 addresses.</p> <p>Specifying the ipv6 address <i>ipv6-prefix/prefix-length</i> command with the eui-64 keyword configures site-local and global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID.</p> <p>The link-local address for an interface is automatically configured when IPv6 is enabled on the interface.</p>
	Router(config-if)# ipv6 address <i>ipv6-address</i> {/prefix-length link-local }	<p>Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface.</p> <p>Specifying the ipv6 address <i>ipv6-address</i> interface configuration command without the link-local keyword configures site-local and global IPv6 addresses. (The link-local address for an interface is automatically configured when IPv6 is enabled on that interface.)</p> <p>Specifying the ipv6 address command with the link-local keyword configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.</p>
	Router(config-if)# ipv6 unnumbered <i>interface-type</i> <i>interface-number</i>	Specifies an unnumbered interface and enables IPv6 processing on an interface. The global IPv6 address of the interface specified with the <i>interface-type interface-number</i> argument is used as the source address in packets generated from the unnumbered interface. (A link-local address is automatically configured on an unnumbered interface when IPv6 is enabled on the interface.)
	Router(config-if)# ipv6 enable	Automatically configures an IPv6 link-local address on the interface while also enabling the interface for IPv6 processing. The link-local address can be used only to communicate with nodes on the same link.

**Note**

The *ipv6-address* argument in the **ipv6 address** command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

The *ipv6-prefix* argument in the **ipv6 address** command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

The *prefix-length* argument in the **ipv6 address** command is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.

**Note**

In Cisco IOS Release 12.2(4)T or later releases and Cisco IOS Release 12.0(21)ST or later releases, the **ipv6 address** or **ipv6 address eui-64** command can be used to configure multiple IPv6 global and site-local addresses within the same prefix on an interface. Multiple IPv6 link-local addresses on an interface are not supported.

Prior to Cisco IOS Releases 12.2(4)T and 12.0(21)ST, the Cisco IOS command-line interface (CLI) displays the following error message when multiple IPv6 addresses within the same prefix on an interface are configured:

```
Prefix <prefix-number> already assigned to <interface-type>
```

See the [“IPv6 Routing and IPv6 Address Configuration Example”](#) section for a configuration example of multiple IPv6 global and site-local addresses within the same prefix on an interface.

Verifying IPv6 Operation and Address Configuration

To verify that IPv6 processing of packets is enabled globally on the router and on applicable interfaces, and that an IPv6 address is configured on applicable interfaces, enter the **show running-config EXEC** command:

```
Router# show running-config

Building configuration...

Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
!
hostname cat
!
ipv6 unicast-routing
!
interface Ethernet0
  no ip route-cache
  no ip mroute-cache
  no keepalive
  media-type 10BaseT
  ipv6 address 3FFE:C00:0:1::/64 eui-64
!
```

**Note**

Display text was omitted from the example.

To verify that IPv6 addresses are configured correctly, enter the **show ipv6 interface EXEC** command. The following example shows the IPv6 addresses configured for Ethernet interface 0:

```
Router# show ipv6 interface ethernet 0

Ethernet0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::260:3EFF:FE11:6770
Global unicast address(es):
  3FFE:C00:0:1:260:3EFF:FE11:6770, subnet is 3FFE:C00:0:1::/64
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF11:6770
MTU is 1500 bytes
ICMP error messages limited to one every 500 milliseconds
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```


Note

For a description of each output display field, refer to the **show ipv6 interface** command in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.

IPv6 Routing and IPv6 Address Configuration Example

In the following example, IPv6 is enabled on the router with both a link-local address and a global address based on the IPv6 prefix 3ffe:c00:c18:1::/64. The EUI-64 interface ID is used in the low-order 64 bits of both addresses. Output from the **show ipv6 interface EXEC** command is included to show how the interface ID (260:3EFF:FE47:1530) is appended to the link-local prefix FE80::/64 of Ethernet interface 0.

```
ipv6 unicast-routing

interface ethernet 0
  ipv6 address 3ffe:c00:c18:1::/64 eui-64

Router# show ipv6 interface ethernet 0

Ethernet0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::260:3EFF:FE47:1530
Global unicast address(es):
  3FFE:C00:C18:1:260:3EFF:FE47:1530, subnet is 3FFE:C00:C18:1::/64
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF47:1530
  FF02::9
MTU is 1500 bytes
ICMP error messages limited to one every 500 milliseconds
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

In the following example, multiple IPv6 global addresses within the prefix 3000::/64 are configured on Ethernet interface 0:

```
interface ethernet 0
  ipv6 address 3000::1/64
  ipv6 address 3000::/64 eui-64
```



Note

All site-local addresses must be within the same site.

Configuring IPv6 ICMP Rate Limiting

In Cisco IOS release 12.2(8)T or later releases, IPv6 Internet Control Message Protocol (ICMP) Rate Limiting implements a token bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent out on the network. The initial implementation of IPv6 ICMP rate limiting defined a fixed interval between error messages, but some applications, such as traceroute, often require replies to a group of requests sent in rapid succession. The fixed interval between error messages is not flexible enough to work with applications such as traceroute and can cause the application to fail. Implementing a token bucket scheme allows a number of tokens—representing the ability to send one error message each—to be stored in a virtual bucket. The maximum number of tokens allowed in the bucket can be specified, and for every error message to be sent, one token is removed from the bucket. If a series of error messages is generated, error messages can be sent until the bucket is empty. When the bucket is empty of tokens, IPv6 ICMP error messages are not sent until a new token is placed in the bucket. The token bucket algorithm does not increase the average rate limiting time interval, and it is more flexible than the fixed time interval scheme.

The following sections explain how to configure the IPv6 ICMP Rate Limiting feature. Each task in the list is identified as either required or optional.

- [Configuring the Rate Limiting Interval and Token Bucket](#) (optional)
- [Verifying IPv6 ICMP Rate Limiting Configuration](#) (optional)

See the “[IPv6 ICMP Rate Limiting Configuration Example](#)” section for a configuration example.

Configuring the Rate Limiting Interval and Token Bucket

To configure the interval and bucket size for IPv6 ICMP error messages, use the following command in global configuration mode:

Command	Purpose
Router(config)# ipv6 icmp error-interval <i>milliseconds</i> [<i>bucketsize</i>]	Configures the interval and bucket size for IPv6 ICMP error messages. The <i>milliseconds</i> argument specifies the interval between tokens being added to the bucket. The optional <i>bucketsize</i> argument defines the maximum number of tokens stored in the bucket.

Verifying IPv6 ICMP Rate Limiting Configuration

Enter the **show ipv6 traffic EXEC** command to display ICMP rate-limited counters:

```
Router# show ipv6 traffic
```

```
ICMP statistics:
  Rcvd: 188 input, 0 checksum errors, 0 too short
        0 unknown info type, 0 unknown error type
  unreachable: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
  parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout, 0 too big
        0 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        1 router solicit, 175 router advert, 0 redirects
        0 neighbor solicit, 12 neighbor advert
  Sent: 7376 output, 56 rate-limited
        unreachable: 0 routing, 15 admin, 0 neighbor, 0 address, 0 port
        parameter: 0 error, 0 header, 0 option
        0 hopcount expired, 0 reassembly timeout, 0 too big
        15 echo request, 0 echo reply
        0 group query, 0 group report, 0 group reduce
        0 router solicit, 7326 router advert, 0 redirects
        2 neighbor solicit, 22 neighbor advert
```



Note

For a description of each output display field, refer to the **show ipv6 traffic** command in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.

IPv6 ICMP Rate Limiting Configuration Example

The following example shows an interval of 50 milliseconds and a bucket size of 20 tokens being configured for IPv6 ICMP error messages:

```
ipv6 icmp error-interval 50 20
```

Configuring Static Cache Entries for IPv6 Neighbor Discovery

The tasks in the following sections explain how to configure the Static Cache Entry for IPv6 Neighbor Discovery feature. Each task in the list is identified as either required or optional:

- [Configuring a Static Cache Entry](#) (required)
- [Verifying the Static Cache Entry Configuration](#) (optional)

See the “[Static Cache Entry for IPv6 Neighbor Discovery Configuration Example](#)” section for a configuration example.



Note

Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

Configuring a Static Cache Entry

The Cisco IOS software uses static entries in the IPv6 neighbor discovery cache to translate 128-bit IPv6 addresses into 48-bit hardware addresses (functionality in IPv6 that is equivalent to static Address Resolution Protocol (ARP) entries in IPv4, which are used to translate 32-bit IP addresses into 48-bit hardware addresses).

To configure a static entry in the IPv6 neighbor discovery cache, use the following command in global configuration mode:

Command	Purpose
<pre>Router(config)# ipv6 neighbor ipv6-address interface-type interface-number hardware-address</pre>	<p>Configures a static entry in the IPv6 neighbor discovery cache.</p> <p>The <i>ipv6-address</i> argument specifies the IPv6 address that corresponds to the local data-link address.</p> <p>The <i>interface-type</i> argument specifies the interface type. For supported interface types, use the question mark (?) online help function.</p> <p>The <i>interface-number</i> argument specifies the interface number.</p> <p>The <i>hardware-address</i> argument specifies the local data-link address (a 48-bit address).</p>

Verifying the Static Cache Entry Configuration

To display IPv6 neighbor discovery cache information, use the **show ipv6 neighbors EXEC** command. A hyphen (-) in the Age field of the command output indicates a static entry. The following example displays IPv6 neighbor discovery cache information for Ethernet interface 2:

```
Router# show ipv6 neighbors ethernet 2

IPv6 Address                               Age Link-layer Addr State Interface
2000:0:0:4::2                               0 0003.a0d6.141e REACH Ethernet2
FE80::203:A0FF:FED6:141E                     0 0003.a0d6.141e REACH Ethernet2
3001:1::45a                                 - 0002.7d1a.9472 REACH Ethernet2
```



Note

For a description of each output display field, refer to the **show ipv6 neighbors** command in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.

Static Cache Entry for IPv6 Neighbor Discovery Configuration Example

The following example configures a static entry in the IPv6 neighbor discovery cache for a neighbor with the IPv6 address 3001:001::45a and link-layer address 0002.7d1a.9472 on Ethernet interface 1:

```
ipv6 neighbor 3001:001::45a ethernet1 0002.7d1a.9472
```

Configuring IPv6 Duplicate Address Detection

The IPv6 Duplicate Address Detection feature verifies the uniqueness of new unicast IPv6 addresses before assigning the addresses to interfaces (the new addresses remain in a tentative state while duplicate address detection is performed). Duplicate address detection uses neighbor solicitation messages to verify the uniqueness of unicast IPv6 addresses.

There are no required configuration tasks for the IPv6 Duplicate Address Detection feature; duplicate address detection on tentative unicast IPv6 addresses and the automatic sending of consecutive neighbor solicitation messages during duplicate address detection are enabled by default. However, the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is performed on tentative unicast IPv6 addresses can be configured.

The tasks in the following sections explain how to configure the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is performed on tentative unicast IPv6 addresses. Each task in the list is identified as either required or optional:

- [Configuring Duplicate Address Detection Neighbor Solicitation Messages](#) (optional)
- [Verifying the Duplicate Address Detection Neighbor Solicitation Messages Configuration](#) (optional)

See the “[IPv6 Duplicate Address Detection Configuration Example](#)” section for a configuration example.



Note

Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

Configuring Duplicate Address Detection Neighbor Solicitation Messages

To configure the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is performed on tentative unicast IPv6 addresses, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# <code>ipv6 nd dad attempts value</code>	<p>Configures the number of consecutive neighbor solicitation messages that are sent on the specified IPv6 interface while duplicate address detection is performed on the tentative unicast IPv6 address of the interface.</p> <p>The <i>value</i> argument specifies the number of messages. The acceptable range is from 0 to 600. Configuring a value of 0 disables duplicate address detection on the specified interface; a value of 1 (the default value) configures a single transmission without follow-up transmissions.</p> <p>The no form of the <code>ipv6 nd dad attempts</code> command returns the number of messages that are sent on the specified interface to the default value (1).</p>

Verifying the Duplicate Address Detection Neighbor Solicitation Messages Configuration

To view the state (OK, TENTATIVE, or DUPLICATE) of the unicast IPv6 addresses configured for an interface, to verify whether duplicate address detection is enabled on the interface, and to verify the number of consecutive duplicate address detection, neighbor solicitation messages that are being sent on the interface, enter the **show ipv6 interface EXEC** command. The following example shows duplicate address detection specifics for Ethernet interface 0:

```
Router# show ipv6 interface ethernet 0

Ethernet0 is up, line protocol is up
 IPv6 is stalled, link-local address is FE80::1 [TENTATIVE]
 Global unicast address(es):
   2000::1, subnet is 2000::/64 [TENTATIVE]
   3000::1, subnet is 3000::/64 [TENTATIVE]
 Joined group address(es):
   FF02::1
   FF02::2
   FF02::1:FF00:1
 MTU is 1500 bytes
 ICMP error messages limited to one every 100 milliseconds
 ICMP redirects are enabled
 ND DAD is enabled, number of DAD attempts: 1
 ND reachable time is 30000 milliseconds
 ND advertised reachable time is 0 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND router advertisements are sent every 200 seconds
 ND router advertisements live for 1800 seconds
 Hosts use stateless autoconfig for addresses.
```



Note

For a description of each output display field, refer to the **show ipv6 interface** command in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.

Duplicate address detection is suspended on interfaces that are administratively “down.” While an interface is administratively “down,” the unicast IPv6 addresses assigned to the interface are set to a pending state. Duplicate address detection is automatically restarted on an interface when the interface returns to being administratively “up.”



Note

An interface returning to administratively “up” restarts duplicate address detection for all of the unicast IPv6 addresses on the interface. While duplicate address detection is performed on the link-local address of an interface, the state for the other IPv6 addresses is still set to TENTATIVE. When duplicate address detection is completed on the link-local address, duplicate address detection is performed on any remaining IPv6 addresses that were not derived from the interface identifier of the link-local address.

When duplicate address detection identifies a duplicate address, the state of the address is set to DUPLICATE and the address is not used, as shown in the following sample output from the **show ipv6 interface EXEC** command:

```
Router# show ipv6 interface ethernet 0

Ethernet0 is up, line protocol is up
 IPv6 is stalled, link-local address is FE80::1 [DUPLICATE]
 Global unicast address(es):
   2000::1, subnet is 2000::/64 [TENTATIVE]
```

```

3000::1, subnet is 3000::/64 [TENTATIVE]
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF00:1
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.

```

If the duplicate address is the link-local address of the interface, as shown in the sample output, the processing of IPv6 packets is disabled on the interface and an error message similar to the following is issued:

```
%IPv6-4-DUPLICATE: Duplicate address FE80::1 on Ethernet0
```

If the duplicate address is a global address of the interface, the address is not used and an error message similar to the following is issued:

```
%IPv6-4-DUPLICATE: Duplicate address 3000::/64 on Ethernet0
```

IPv6 Duplicate Address Detection Configuration Example

The following example configures five consecutive neighbor solicitation messages to be sent on Ethernet interface 0 while duplicate address detection is being performed on the tentative unicast IPv6 address of the interface. The example also disables duplicate address detection processing on Ethernet interface 1.

```

interface ethernet 0
  ipv6 nd dad attempts 5

interface ethernet 1
  ipv6 nd dad attempts 0

```



Note

Configuring a value of 0 with the **ipv6 nd dad attempts** interface configuration command disables duplicate address detection processing on the specified interface; a value of 1 configures a single transmission without follow-up transmissions. The default is one message.

Configuring IPv6 Redirect Messages

The IPv6 Redirect Messages feature enables a router to send Internet Control Message Protocol (ICMP) IPv6 neighbor redirect messages to inform hosts of better first hop nodes (routers or hosts) on the path to a destination.

There are no configuration tasks for the IPv6 Redirect Messages feature. The sending of IPv6 redirect messages is enabled by default. Use the **no ipv6 redirects** interface configuration command to disable the sending of IPv6 redirect messages on an interface. Use the **ipv6 redirects** interface configuration command to reenables the sending of IPv6 redirect messages if the Cisco IOS software is forced to resend a packet through the same interface on which the packet was received.

To verify whether the sending of IPv6 redirect messages is enabled on an interface, enter the **show ipv6 interface EXEC** command. The following example shows IPv6 redirect messages for Ethernet interface 0:

```
Router# show ipv6 interface ethernet 0

Ethernet0 is up, line protocol is up
  IPv6 is stalled, link-local address is FE80::1
  Global unicast address(es):
    2000::1, subnet is 2000::/64
    3000::1, subnet is 3000::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

**Note**

For a description of each output display field, refer to the **show ipv6 interface** command in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.

IPv6 Redirect Messages Configuration Example

The following example disables the sending of ICMP IPv6 redirect messages on Ethernet interface 0 and reenables the messages on Ethernet interface 1:

```
interface ethernet 0
  no ipv6 redirects

interface ethernet 1
  ipv6 redirects
```

Mapping Host Names to IPv6 Addresses

Host names can be associated with IPv6 addresses. The Cisco IOS software maintains a cache of host name-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. This cache speeds the process of converting names to addresses.

Similar to IPv4, IPv6 uses a naming scheme that allows a network device to be identified by its location within a hierarchical name space that provides for domains. Domain names are joined together with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that is identified by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, the File Transfer Protocol (FTP) server, for example, is identified as *ftp.cisco.com*.

A *name server* is used to keep track of information associated with domain names. A name server can maintain a database of host name-to-address mappings. Each name can map to one or more IPv4 addresses, IPv6 addresses, or both address types. In order to use this service to map domain names to IPv6 addresses, you must specify a name server and enable the Domain Naming System (DNS)—the global naming scheme of the Internet that uniquely identifies network devices. The tasks for mapping host names to IPv6 addresses are described in the following sections. Each task in the list is identified as either required or optional:

- [Assigning Host Names to IPv6 Addresses](#) (required)
- [Specifying a Default Domain Name](#) (optional)
- [Specifying a Name Server](#) (required)
- [Enabling the DNS](#) (required)
- [Verifying the Host Name-to-Address Mappings Configuration](#) (optional)

See the “[Host Name-to-Address Mappings Configuration Example](#)” section for a configuration example.



Note

Refer to the “Configuring IP Addressing” chapter of the Release 12.2, *Cisco IOS IP Configuration Guide* for tasks specific to mapping domain names to IPv4 addresses. The following sections contain tasks that are specific to mapping domain names to IPv6 addresses.

Assigning Host Names to IPv6 Addresses

Typically, it is easier to refer to network devices by symbolic names rather than numerical addresses (services such as Telnet can use host names or addresses). Host names and IPv6 addresses can be associated with one another through static or dynamic means.

Manually assigning host names to addresses is useful when dynamic mapping is not available. To assign host names to IPv6 addresses, use the following command in global configuration mode:

Command	Purpose
Router(config)# ipv6 host name [port] ipv6-address1 [ipv6-address2...ipv6-address4]	Defines a static host name-to-address mapping in the host name cache.

Specifying a Default Domain Name

You can specify a default domain name that the Cisco IOS software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any host name that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up.

To specify a domain name or names, use either of the following commands in global configuration mode:

**Note**

The following commands are used to specify default domain names that can be used by both IPv4 and IPv6.

Command	Purpose
Router(config)# ip domain-name <i>name</i>	Defines a default domain name that the Cisco IOS software will use to complete unqualified host names.
Router(config)# ip domain-list <i>name</i>	Defines a list of default domain names to complete unqualified host names.

Specifying a Name Server

To specify one or more hosts (up to six) that can function as a name server to supply name information for the DNS, use the following command in global configuration mode:

Command	Purpose
Router(config)# ip name-server <i>server-address1</i> [<i>server-address2</i> ... <i>server-address6</i>]	Specifies one or more hosts that supply name information. Note The <i>server-address</i> argument can be either an IPv4 or IPv6 address.

Enabling the DNS

To reenable DNS if it has been disabled (DNS is enabled by default), use the following command in global configuration mode:

Command	Purpose
Router(config)# ip domain-lookup	Enables DNS-based host name-to-address translation.

Verifying the Host Name-to-Address Mappings Configuration

To verify static host name-to-address mappings, default domain names, and name servers in the host name cache, and to verify that the DNS service is enabled, enter the **show running-config EXEC** command:

```
Router# show running-config

Building configuration...
!
ipv6 host cisco-sj 3FFE:700:20:1::12
!
ip domain-name cisco.com
ip domain-lookup
ip name-server 2002:C01F:768::1
```

**Note**

Display text was omitted from the example.

Host Name-to-Address Mappings Configuration Example

The following example defines two static host name-to-address mappings in the host name cache, establishes a domain list with several alternate domain names to complete unqualified host names, specifies host 3FFE:C00::250:8BFF:FEE8:F800 and host 3FFE:80A0:0:F004::1 as the name servers, and reenables the DNS service:

```
ipv6 host cisco-sj 3FFE:700:20:1::12
ipv6 host cisco-hq 2002:C01F:768::1 3FFE:700:20:1::22
ip domain-list csi.com
ip domain-list telecomprog.edu
ip domain-list merit.edu
ip name-server 3FFE:C00::250:8BFF:FEE8:F800 3FFE:80A0:0:F004::1
ip domain-lookup
```

Configuring IPv6 ATM and Frame Relay Interfaces

IPv6 for Cisco IOS software supports wide-area networking technologies such as Cisco High-Level Data Link Control (HDLC), PPP over Packet-Over-SONET, ISDN, and serial (synchronous and asynchronous) interface types, ATM permanent virtual circuits (PVCs), and Frame Relay PVCs. These technologies function the same in IPv6 as they do in IPv4—IPv6 does not enhance the technologies in any way. However, new commands for mapping protocol (network-layer) addresses to ATM and Frame Relay PVCs have been introduced for IPv6.

The following sections describe how to map IPv6 addresses to ATM and Frame Relay PVCs.



Note

The following sections are not applicable to ATM LAN Emulation (LANE).

For a complete description of the commands that appear in the following sections, refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.



Note

The following sections do not provide in-depth information on customizing ATM and Frame Relay because the technologies function the same in IPv6 as they do in IPv4. Refer to the “Configuring ATM” and “Configuring Frame Relay” chapters of the Release 12.2, *Cisco IOS Wide-Area Networking Configuration Guide* publication for information on customizing ATM and Frame Relay. Wherever applicable, the “Usage Guidelines” section of each command reference page in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document lists the equivalent IPv4 ATM and Frame Relay command for each new IPv6 ATM and Frame Relay command. For a complete description of the IPv4 ATM and Frame Relay commands available in the Cisco IOS software, refer to the “ATM Commands” and “Frame Relay Commands” chapters of the Release 12.2, *Cisco IOS Wide-Area Networking Command Reference*. To locate documentation for all other IPv4 commands, use the command reference master index or search online.

Broadcast and multicast are used in LANs to map protocol (network-layer) addresses to the hardware addresses of remote nodes (hosts and routers). Because using broadcast and multicast to map network-layer addresses to hardware addresses in circuit-based WANs, such as ATM and Frame Relay networks, is difficult to implement, these networks utilize implicit, explicit, and dynamic mappings for the network-layer addresses of remote nodes and the PVCs used to reach the addresses.

Assigning an IPv6 address to an interface by using the **ipv6 address** interface configuration command defines the IPv6 addresses for the interface and the network that is directly connected to the interface. If only one PVC is terminated on the interface (the interface is a point-to-point interface), there is an implicit mapping between all of the IPv6 addresses on the network and the PVC used to reach the addresses (no additional address mappings are needed). If several PVCs are terminated on the interface (the interface is a point-to-multipoint interface), the **protocol ipv6** ATM VC configuration command (for ATM networks) or the **frame-relay map ipv6** interface configuration command (for Frame Relay networks) is used to configure explicit mappings between the IPv6 addresses of the remote nodes and the PVCs used to reach the addresses.

**Note**

Given that IPv6 supports multiple address types and depending on which applications or protocols are configured on a point-to-multipoint interface, you may need to configure multiple explicit mappings between the IPv6 addresses of the interface and the PVC used to reach the addresses. For example, explicitly mapping both the link-local and global IPv6 address of a point-to-multipoint interface to the PVC that the interface terminates ensures that the Interior Gateway Protocol (IGP) configured on the interface forwards traffic to and from the PVC correctly.

In IPv4, the Inverse Address Resolution Protocol (Inverse ARP) is used to dynamically map the network-layer address of a node at the far end of a PVC to that PVC. Refer to the “Configuring ATM” and “Configuring Frame Relay” chapters of the Release 12.2, *Cisco IOS Wide-Area Networking Configuration Guide* publication for information on configuring Inverse ARP. In IPv6, Inverse Neighbor Discovery is used to dynamically map the global IPv6 address of a node at the far end of a PVC to that PVC.

**Note**

Support for Inverse Neighbor Discovery is not in the current releases of the Cisco IOS software. Inverse Neighbor Discovery will be supported in a later release of the Cisco IOS software. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features.

The tasks in the following sections explain how to explicitly map IPv6 addresses to the ATM and Frame Relay PVCs used to reach the addresses. Each task in the list is identified as either required or optional.

- [Mapping an IPv6 Address to an ATM PVC](#) (required)
- [Mapping an IPv6 Address to a Frame Relay PVC](#) (required)
- [Verifying the IPv6 Address to ATM and Frame Relay PVC Mapping](#) (optional)

See the “[IPv6 Address to ATM and Frame Relay PVC Mapping Configuration Examples](#)” section for configuration examples.

Mapping an IPv6 Address to an ATM PVC

To map the IPv6 address of a remote node to the PVC used to reach the address, use the following command in ATM VC configuration mode:

Command	Purpose
<pre>Router(config-if-atm-vc)# protocol ipv6 ipv6-address [[no] broadcast]</pre>	<p>Maps the IPv6 address of a remote node to the PVC used to reach the address.</p> <p>The <i>ipv6-address</i> argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.</p> <p>The optional [no] broadcast keywords indicate whether the map entry should be used when IPv6 multicast packets (not broadcast packets) are sent to the interface. Pseudobroadcasting is supported. The [no] broadcast keywords in the protocol ipv6 command take precedence over the broadcast command configured on the same ATM PVC.</p>

Mapping an IPv6 Address to a Frame Relay PVC

To map the IPv6 address of a remote node to the data-link connection identifier (DLCI) of the PVC used to reach the address, use the following command in interface configuration mode:

Command	Purpose
<pre>Router(config-if)# frame-relay map ipv6 ipv6-address dlci [broadcast] [cisco] [ietf] [payload-compression]</pre>	<p>Maps the IPv6 address of a remote node to the DLCI of the PVC used to reach the address.</p> <p>The <i>ipv6-address</i> argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.</p> <p>The <i>dlci</i> argument specifies the DLCI number used to reach the specified IPv6 address on the interface. The acceptable range is from 16 to 1007.</p> <p>The optional broadcast keyword specifies that IPv6 multicast packets (not broadcast packets) should be forwarded to this IPv6 address when multicast is not enabled. (Refer to the frame-relay multicast-dlci command in the “Frame Relay Commands” chapter of the Release 12.2, <i>Cisco IOS Wide-Area Networking Command Reference</i> for information on defining a multicast DLCI.)</p> <p>The optional cisco keyword specifies the Cisco form of Frame Relay encapsulation.</p> <p>The optional ietf keyword specifies the Internet Engineering Task Force (IETF) form of Frame Relay encapsulation.</p> <p>The optional payload-compression keyword specifies payload compression. (Refer to the frame-relay map ipv6 command in the <i>IPv6 for Cisco IOS Software, File 3 of 3: Commands</i> document for subordinate keywords and arguments related to the payload-compression keyword.)</p>

Verifying the IPv6 Address to ATM and Frame Relay PVC Mapping

To verify that the IPv6 address of a remote node is mapped to the PVC used to reach the address, use the **show atm map** privileged EXEC command. The following example shows that the link-local and global IPv6 addresses (FE80::60:3E47:AC8:C and 3ffe:1111:2222:1003::72, respectively) of a remote node are explicitly mapped to PVC 1/32 of ATM interface 0.

```
Router# show atm map
```

```
Map list ATM0pvc1 : PERMANENT
ipv6 FE80::60:3E47:AC8:C maps to VC 1, VPI 1, VCI 32, ATM0
, broadcast
ipv6 3ffe:1111:2222:1003::72 maps to VC 1, VPI 1, VCI 32, ATM0
```

To verify that the IPv6 address of a remote node is mapped to the DLCI of the PVC used to reach the address, enter the **show frame-relay map** EXEC command. The following example shows that the link-local and global IPv6 addresses (FE80::E0:F727:E400:A and 3ffe:1111:2222:1044::73; FE80::60:3E47:AC8:8 and 3ffe:1111:2222:1044::72) of two remote nodes are explicitly mapped to DLCI 17 and DLCI 19, respectively. Both DLCI 17 and DLCI 19 are terminated on interface serial 3 of this node; therefore, interface serial 3 of this node is a point-to-multipoint interface.

```
Router# show frame-relay map
```

```
Serial3 (up): ipv6 FE80::E0:F727:E400:A dlci 17(0x11,0x410), static,
broadcast, CISCO, status defined, active
Serial3 (up): ipv6 3ffe:1111:2222:1044::72 dlci 19(0x13,0x430), static,
CISCO, status defined, active
Serial3 (up): ipv6 3ffe:1111:2222:1044::73 dlci 17(0x11,0x410), static,
CISCO, status defined, active
Serial3 (up): ipv6 FE80::60:3E47:AC8:8 dlci 19(0x13,0x430), static,
broadcast, CISCO, status defined, active
```

IPv6 Address to ATM and Frame Relay PVC Mapping Configuration Examples

This section provides the following IPv6 ATM and Frame Relay PVC mapping configuration examples:

- [IPv6 ATM PVC Mapping Configuration Example—Point-to-Point Interface](#)
- [IPv6 ATM PVC Mapping Configuration Example—Point-to-Multipoint Interface](#)
- [IPv6 Frame Relay PVC Mapping Configuration Example—Point-to-Point Interface](#)
- [IPv6 Frame Relay PVC Mapping Configuration Example—Point-to-Multipoint Interface](#)

IPv6 ATM PVC Mapping Configuration Example—Point-to-Point Interface

In the following example, two nodes named Cisco 1 and Cisco 2 are connected by a single PVC. The point-to-point subinterface ATM0.132 is used on both nodes to terminate the PVC; therefore, the mapping between the IPv6 addresses of both nodes and the PVC is implicit (no additional mappings are required).

Cisco 1 Configuration

```
interface ATM0
 no ip address
!
interface ATM0.132 point-to-point
 pvc 1/32
 encapsulation aal5snap
!
 ipv6 address 3ffe:1111:2222:1003::72/64
```

Cisco 2 Configuration

```
interface ATM0
  no ip address
  !
interface ATM0.132 point-to-point
  pvc 1/32
    encapsulation aal5snap
    !
  ipv6 address 3ffe:1111:2222:1003::45/64
```

IPv6 ATM PVC Mapping Configuration Example—Point-to-Multipoint Interface

In the following example, the same two nodes (Cisco 1 and Cisco 2) from the previous example are connected by the same PVC. In this example, however, the point-to-multipoint interface ATM0 is used on both nodes to terminate the PVC; therefore, explicit mappings are required between the link-local and global IPv6 addresses of interface ATM0 on both nodes and the PVC. Additionally, ATM pseudobroadcasts are enabled on the link-local address of interface ATM0 on both nodes.

Cisco 1 Configuration

```
interface ATM0
  no ip address
  pvc 1/32
    protocol ipv6 3ffe:1111:2222:1003::45
    protocol ipv6 FE80::60:2FA4:8291:2 broadcast
    encapsulation aal5snap
    !
  ipv6 address 3ffe:1111:2222:1003::72/64
```

Cisco 2 Configuration

```
interface ATM0
  no ip address
  pvc 1/32
    protocol ipv6 FE80::60:3E47:AC8:C broadcast
    protocol ipv6 3ffe:1111:2222:1003::72
    encapsulation aal5snap
    !
  ipv6 address 3ffe:1111:2222:1003::45/64
```

IPv6 Frame Relay PVC Mapping Configuration Example—Point-to-Point Interface

In the following example, three nodes named Cisco A, Cisco B, and Cisco C make up a fully meshed network. Each node is configured with two PVCs, which provide an individual connection to each of the other two nodes. Each PVC is configured on a different point-to-point subinterface, which creates three unique IPv6 networks (3ffe:1111:2222:1017:/64, 3ffe:1111:2222:1018:/64, and 3ffe:1111:2222:1019:/64). Therefore, the mappings between the IPv6 addresses of each node and the DLCI (DLCI 17, 18, and 19) of the PVC used to reach the addresses are implicit (no additional mappings are required).

**Note**

Given that each PVC in the following example is configured on a different point-to-point subinterface, the configuration in the following example can also be used in a network that is not fully meshed. Additionally, configuring each PVC on a different point-to-point subinterface can help simplify your routing protocol configuration. However, the configuration in the following example requires more than one IPv6 network, whereas configuring each PVC on point-to-multipoint interfaces requires only one IPv6 network.

Cisco A Configuration

```
interface Serial3
 encapsulation frame-relay
 !
interface Serial3.17 point-to-point
 description to Cisco B
 ipv6 address 3ffe:1111:2222:1017::46/64
 frame-relay interface-dlci 17
 !
interface Serial3.19 point-to-point
 description to Cisco C
 ipv6 address 3ffe:1111:2222:1019::46/64
 frame-relay interface-dlci 19
```

Cisco B Configuration

```
interface Serial5
 encapsulation frame-relay
 !
interface Serial5.17 point-to-point
 description to Cisco A
 ipv6 address 3ffe:1111:2222:1017::73/64
 frame-relay interface-dlci 17
 !
interface Serial5.18 point-to-point
 description to Cisco C
 ipv6 address 3ffe:1111:2222:1018::73/64
 frame-relay interface-dlci 18
```

Cisco C Configuration

```
interface Serial0
 encapsulation frame-relay
 !
interface Serial0.18 point-to-point
 description to Cisco B
 ipv6 address 3ffe:1111:2222:1018::72/64
 frame-relay interface-dlci 18
 !
interface Serial0.19 point-to-point
 description to Cisco A
 ipv6 address 3ffe:1111:2222:1019::72/64
 frame-relay interface-dlci 19
```

IPv6 Frame Relay PVC Mapping Configuration Example—Point-to-Multipoint Interface

In the following example, the same three nodes (Cisco A, Cisco B, and Cisco C) from the previous example make up a fully meshed network and each node is configured with two PVCs (which provide an individual connection to each of the other two nodes). However, the two PVCs on each node in the following example are configured on a single interface (serial 3, serial 5, and serial 10, respectively), which makes each interface a point-to-multipoint interface. Therefore, explicit mappings are required between the link-local and global IPv6 addresses of each interface on all three nodes and the DLCI (DLCI 17, 18, and 19) of the PVC used to reach the addresses.

Cisco A Configuration

```
interface Serial3
 encapsulation frame-relay
 ipv6 address 3ffe:1111:2222:1044::46/64
 frame-relay map ipv6 FE80::E0:F727:E400:A 17 broadcast
 frame-relay map ipv6 FE80::60:3E47:AC8:8 19 broadcast
 frame-relay map ipv6 3ffe:1111:2222:1044::72 19
 frame-relay map ipv6 3ffe:1111:2222:1044::73 17
```

Cisco B Configuration

```
interface Serial5
 encapsulation frame-relay
 ipv6 address 3ffe:1111:2222:1044::73/64
 frame-relay map ipv6 FE80::60:3E59:DA78:C 17 broadcast
 frame-relay map ipv6 FE80::60:3E47:AC8:8 18 broadcast
 frame-relay map ipv6 3ffe:1111:2222:1044::46 17
 frame-relay map ipv6 3ffe:1111:2222:1044::72 18
```

Cisco C Configuration

```
interface Serial0
 encapsulation frame-relay
 ipv6 address 3ffe:1111:2222:1044::72/64
 frame-relay map ipv6 FE80::60:3E59:DA78:C 19 broadcast
 frame-relay map ipv6 FE80::E0:F727:E400:A 18 broadcast
 frame-relay map ipv6 3ffe:1111:2222:1044::46 19
 frame-relay map ipv6 3ffe:1111:2222:1044::73 18
```

Configuring IPv6 Standard Access Lists

An IPv6 standard access list is a sequential collection of permit and deny conditions that apply to IPv6 addresses. The Cisco IOS software tests addresses against the conditions in an access list one by one. The first match determines whether the software accepts or rejects the address. Because the software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the software rejects the address.

IPv6 standard access lists are identified by names and use source and destination IPv6 prefixes for matching operations. IPv6 standard access lists should be defined and then assigned to an interface; otherwise, the interface operates with an empty access list until an access list is defined for the interface.

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

- [Creating and Applying an IPv6 Standard Access List](#) (required)
- [Verifying IPv6 Standard Access List Configuration](#) (optional)

See the [“IPv6 Standard Access List Configuration Example”](#) section for a configuration example.

Creating and Applying an IPv6 Standard Access List

To create an IPv6 standard access list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	<pre>Router(config)# ipv6 access-list access-list-name {permit deny} {source-ipv6-prefix/prefix-length any} {destination-ipv6-prefix/prefix-length any} [priority value]</pre>	<p>Defines a standard IPv6 access list and sets deny or permit conditions for the access list.</p> <p>The <i>access-list name</i> argument specifies the name of the IPv6 access list. Access list names cannot contain a space or quotation mark.</p> <p>The deny keyword specifies deny conditions for the access list.</p> <p>The permit keyword specifies permit conditions for the access list.</p> <p>The <i>source-ipv6-prefix/prefix-length</i> and <i>destination-ipv6-prefix/prefix-length</i> arguments are mandatory. The <i>prefix-length</i> argument indicates the number of consecutive, most significant bits that are used in the match. A slash mark must precede the decimal value.</p> <p>The any keyword, when specified instead of the <i>source-ipv6-prefix/prefix-length</i> or <i>destination-ipv6-prefix/prefix-length</i>, matches any prefix and is equivalent to the IPv6 prefix <code>::/0</code>.</p> <p>The priority keyword specifies the order in which the statement is applied in the access list. The acceptable range is from 1 to 4294967295.</p>
Step 2	<pre>Router(config)# interface interface-type interface-number</pre>	<p>Specifies an interface type and number, and places the router in interface configuration mode.</p>
Step 3	<pre>Router(config-if)# ipv6 traffic-filter access-list-name {in out}</pre>	<p>Applies the specified IPv6 access list to the interface specified in the previous step.</p> <p>The in keyword filters incoming IPv6 traffic on the specified interface.</p> <p>The out keyword filters outgoing IPv6 traffic on the specified interface.</p>



Note

Both the *source-ipv6-prefix/prefix-length* and *destination-ipv6-prefix/prefix-length* arguments are used for traffic filtering (the source prefix filters traffic based upon the traffic source; the destination prefix filters traffic based upon the traffic destination). IPv6 prefix lists, not access lists, should be used for filtering routing protocol prefixes.

The Cisco IOS software compares an IPv6 prefix against the **permit** and **deny** condition statements in the access list. Every IPv6 access list, including access lists that do not have any **permit** and **deny** condition statements, has an implicit **deny any any** statement as its last match condition. The priority value applied to each condition statement dictates the order in which the statement is applied in the access list.

Verifying IPv6 Standard Access List Configuration

To verify that IPv6 standard access lists are configured correctly, enter the **show ipv6 access-list** EXEC command:

```
Router# show ipv6 access-list

ipv6 access-list list1
  deny 3000::/64 any priority 10
  permit 2000::/64 any priority 20
  permit any any priority 30
```



Note

For a description of each output display field, refer to the **show ipv6 access-list** command in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.

To verify that all access lists (both IPv6 and IPv4 access lists) are configured correctly, enter the **show running-config** EXEC command:

```
Router# show running-config

Building configuration...

Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by user1
!
hostname router1
!
ipv6 unicast-routing
!
ip classless
!
access-list 2 permit 10.1.1.2
access-list 198 permit ip 172.16.0.0 0.0.255.255 any
access-list 198 permit ip 192.168.0.0 0.0.255.255 any
!
ipv6 access-list list1 permit 2000::/64 any priority 10
ipv6 access-list list1 deny 3000::/64 any priority 20
ipv6 access-list list1 permit any any priority 30
!
```



Note

Display text was omitted from the example.

IPv6 Standard Access List Configuration Example

The following example configures the access list named list2 and applies the access list to outbound traffic on Ethernet interface 0. Specifically, the first access list entry keeps all packets from the network fec0:0:0:2::/64 (packets that have the site-local prefix fec0:0:0:2 as the first 64 bits of their source IPv6 address) from exiting out of Ethernet interface 0. The second entry in the access list permits all other traffic to exit out of Ethernet interface 0. The second entry is necessary because an implicit deny all condition is at the end of each IPv6 access list.

```
ipv6 access-list list2 deny fec0:0:0:2::/64 any
ipv6 access-list list2 permit any any

interface ethernet 0
  ipv6 traffic-filter dublin out
```

Configuring IPv6 RIP

This section describes how to configure Routing Information Protocol (RIP) for IPv6. For a complete description of the IPv6 RIP commands that appear in this section, refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com. For a complete description of the IPv4 RIP commands available in the Cisco IOS software, refer to the RIP Commands chapter of the Release 12.2, *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*. To locate documentation for all other IPv4 commands, use the command reference master index or search online.

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.

RIP in IPv6 functions the same and offers the same benefits as RIP in IPv4. IPv6 enhancements to RIP 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 CLI.



Note

The following sections describe the configuration tasks for creating an IPv6 RIP routing process and enable 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 “Configuring Routing Information Protocol” chapter of the Release 12.2, *Cisco IOS IP Configuration Guide* publication for information on customizing RIP. Wherever applicable, the “Usage Guidelines” section of each command reference page in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document lists the equivalent IPv4 RIP command for each new IPv6 RIP command. Refer to the “RIP Commands” chapter of the Release 12.2, *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols* publication as you customize your IPv6 RIP routing process. Additionally, refer to the “Configuring IP Routing Protocol-Independent Features” chapter of the Release 12.2, *Cisco IOS IP Configuration Guide* publication for protocol-independent features that also apply to RIP.

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

- [Enabling IPv6 RIP](#) (required)
- [Originating a Default IPv6 Route into an IPv6 RIP Routing Process](#) (optional)
- [Redistributing Routes into an IPv6 RIP Routing Process](#) (optional)
- [Filtering IPv6 RIP Routing Updates](#) (optional)
- [Verifying IPv6 RIP Configuration and Operation](#) (optional)

See the “[IPv6 RIP Configuration Example](#)” section for a configuration example.

Enabling IPv6 RIP

To enable IPv6 RIP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ipv6 router rip word Router(config-rtr-rip)# exit	Configures an IPv6 RIP routing process, and places the router in router configuration mode for the IPv6 RIP routing process. Use the exit command to return the router to global configuration mode, from which additional configuration commands are entered.
Step 2	Router(config)# interface interface-type interface-number Router(config-if)#	Specifies the interface type and number, and places the router in interface configuration mode.
Step 3	Router(config-if)# ipv6 rip word enable	Enables the specified IPv6 RIP routing process on an interface.

Originating a Default IPv6 Route into an IPv6 RIP Routing Process

To originate the IPv6 default route (::/0) into a RIP routing process on an interface and to include the default route in router updates sent out of the interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface interface-type interface-number	Specifies the interface type and number, and places the router in interface configuration mode.
Step 2	Router(config-if)# ipv6 rip word default-information {only originate}	<p>Originates the IPv6 default route (::/0) into the specified RIP routing process and includes the default route in router updates sent out of the specified interface.</p> <p>Note To avoid routing loops after the IPv6 default route (::/0) is originated into a specified RIP routing process, the routing process ignores all default route information received in subsequent IPv6 RIP update messages.</p> <p>Specifying the only keyword originates the default route (::/0) and suppresses advertising any routes except the default route sent on this interface.</p> <p>Specifying the originate keyword originates the default route (::/0) and advertises the default route with all other routes in router updates sent on this interface.</p>

Redistributing Routes into an IPv6 RIP Routing Process

To redistribute routes into an IPv6 RIP routing process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router# ipv6 router rip word Router(config-rtr-rip)#	Places the router in router configuration mode for the specified IPv6 RIP routing process.
Step 2	Router(config-rtr-rip)# redistribute protocol [process-id] {level-1 level-1-2 level-2} [metric metric-value] [metric-type {internal external}] [route-map map-name]	<p>Redistributes the specified routes into the IPv6 RIP routing process.</p> <p>The <i>protocol</i> argument can be one of the following keywords: bgp, connected, isis, rip, or static.</p> <p>The rip keyword and <i>process-id</i> argument specify an IPv6 RIP routing process.</p> <p>Note The connected keyword refers to routes that are established automatically by IPv6 having been enabled on an interface.</p>

Filtering IPv6 RIP Routing Updates

To apply a prefix list to IPv6 RIP routing updates that are received or sent on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router# ipv6 router rip word Router(config-rtr-rip)#	Places the router in router configuration mode for the specified IPv6 RIP routing process.
Step 2	Router(config-rtr-rip)# distribute-list prefix-list word {in out} [interface-type interface-number]	<p>Applies the specified prefix list to IPv6 RIP routing updates that are received or sent on the specified interface.</p> <p>Note If an interface is not specified with the distribute-list prefix-list command, the specified prefix list is applied to IPv6 routing updates that are sent or received on all interfaces in the router running the specified process.</p> <p>The in keyword applies the prefix list to incoming routing updates on the specified interface.</p> <p>The out keyword applies the prefix list to outgoing routing updates on the specified interface.</p> <p>Note Redistributed routing information should always be filtered by the distribute-list prefix-list (IPv6 RIP) router configuration command.</p>

Verifying IPv6 RIP Configuration and Operation

To display information about current IPv6 RIP processes, enter the **show ipv6 rip** EXEC command:

```
Router# show ipv6 rip

RIP process "6bone", port 521, multicast-group FF02::9, pid 70
  Administrative distance is 120. Routing table is 0
  Updates every 30 seconds, expire after 180
  Holddown lasts 180 seconds, garbage collect after 120
  Split horizon is on; poison reverse is off
  Default routes are not generated
  Periodic updates 16136, trigger updates 0
```



Note For a description of each output display field, refer to the **show ipv6 rip** command in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.

To display debug messages for IPv6 RIP routing transactions, enter the **debug ipv6 rip** privileged EXEC command:



Note By default, the network server sends the output from debug commands and system error messages to the console. To redirect debug output, use the **logging** command options within configuration mode. Possible destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server. For complete information on debug commands and redirecting debug output, refer to the Release 12.2, *Cisco IOS Debug Command Reference*.

```
Router# debug ipv6 rip

13:09:10:RIPng:Sending multicast update on Ethernet1/1 for as1_rip
13:09:10:      src=FE80::203:E4FF:FE12:CC1D
13:09:10:      dst=FF02::9 (Ethernet1/1)
13:09:10:      sport=521, dport=521, length=32
13:09:10:      command=2, version=1, mbz=0, #rte=1
13:09:10:      tag=0, metric=1, prefix=::/0
13:09:28:RIPng:response received from FE80::202:FDFE:FE77:1E42 on Ethernet1/1 for as1_rip
13:09:28:      src=FE80::202:FDFE:FE77:1E42 (Ethernet1/1)
13:09:28:      dst=FF02::9
13:09:28:      sport=521, dport=521, length=32
13:09:28:      command=2, version=1, mbz=0, #rte=1
13:09:28:      tag=0, metric=1, prefix=2000:0:0:1:1::/80
```

IPv6 RIP Configuration Example

In the following example, the IPv6 RIP process named cisco is enabled on the router and on Ethernet interface 0. The IPv6 default route (::/0) is advertised with all other routes in router updates sent on Ethernet interface 0. Additionally, BGP routes are redistributed into the process cisco and the prefix list named list3 filters inbound routing updates on Ethernet interface 0.

```
ipv6 router rip cisco
  distribute-list prefix-list list3 in ethernet 0
  default-information originate
  redistribute bgp
```

```

interface ethernet 0
  ipv6 address 3ffe:c00:c18:1::/64 eui-64
  ipv6 rip cisco enable

ipv6 prefix-list list3 seq 10 deny ::/0
ipv6 prefix-list list3 seq 15 permit ::/0 le 128

```

Configuring IPv6 IS-IS

This section describes how to configure Intermediate System-to-Intermediate System (IS-IS) for IPv6. In Cisco IOS Release 12.0(21)ST or later releases and Cisco IOS Release 12.2(8)T or later releases, IS-IS is implemented as a supported Interior Gateway Protocol (IGP) for IPv6. Refer to the *Start Here: Cisco IOS Software Release Specifics for IPv6 Features* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for Cisco IOS software release specifics for supported IPv6 features. For a complete description of the IPv6 IS-IS commands that appear in this section, refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in the New Features in Release 12.2 T and the New Features in Release 12.0 ST area of Cisco.com. For a complete description of the IPv4 IS-IS commands available in the Cisco IOS software, refer to the “Integrated IS-IS Commands” chapter of the Release 12.2, *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*. To locate documentation for all other IPv4 commands, use the command reference master index or search online.

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.

IS-IS in IPv6 functions the same and offers many of the same benefits as IS-IS in IPv4. IPv6 enhancements to IS-IS allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and Open System Interconnection (OSI) routes. Extensions to the IS-IS CLI allow configuration of IPv6-specific parameters. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to OSI and IPv4.



Note

The following sections describe the configuration tasks for creating an IPv6 IS-IS routing process and configuring IPv6-specific commands under address family IPv6. The following sections do not provide in-depth information on customizing IS-IS because the protocol functions the same in IPv6 as it does in IPv4. Refer to the “Configuring Integrated IS-IS” chapter of the Release 12.2, *Cisco IOS IP Configuration Guide* publication for information on customizing IS-IS. Wherever applicable, the “Usage Guidelines” section of each command reference page in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document lists the equivalent IPv4 IS-IS command for each new IPv6 IS-IS command. Refer to the “Integrated IS-IS Commands” chapter of the Release 12.2, *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols* publication as you customize your IPv6 IS-IS routing process. Refer also to the “IOS CLNS Commands” chapter of the Release 12.2, *Cisco IOS Apollo Domain, Banyan VINES, DECnet, IOS CLNS, XNS Command Reference* publication for details of Connectionless Network Service (CLNS) commands as you customize your IPv6 IS-IS routing process. Additionally, refer to the “Configuring IP Routing Protocol-Independent Features” chapter of the Release 12.2, *Cisco IOS IP Configuration Guide* publication for protocol-independent features that also apply to IS-IS.

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

- [Configuring IS-IS](#) (required)
- [Assigning an IPv6 IS-IS Routing Process to an Interface](#) (required)
- [Configuring Administrative Distance for IPv6 IS-IS](#) (optional)
- [Configuring the Maximum Number of Parallel Routes for IPv6 IS-IS](#) (optional)
- [Configuring Summary Prefixes for IPv6 IS-IS](#) (optional)
- [Disabling IPv6 Protocol-Support Consistency Checks](#) (optional)
- [Originating a Default IPv6 Route](#) (optional)
- [Redistributing Routes into an IPv6 IS-IS Routing Process](#) (optional)
- [Redistributing IPv6 IS-IS Routes Between IS-IS Levels](#) (optional)
- [Verifying IPv6 IS-IS](#) (optional)

See the “[IPv6 IS-IS Configuration Examples](#)” section for configuration examples.

Configuring IS-IS

Configuring IS-IS comprises two tasks. The first task creates an IS-IS routing process and is performed using protocol-independent IS-IS commands. To create an IS-IS routing process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router isis <i>area-name</i>	<p>Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.</p> <p>The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.</p> <p>The first IS-IS instance is configured Level 1-2 by default. Later instances of IS-IS are automatically Level 1. You can change the level of routing to be performed by a specified routing process using the is-type command.</p>
Step 2	Router(config-router)# net <i>network-entity-title</i>	<p>Configures an IS-IS network entity title (NET) for the routing process.</p> <p>The <i>network-entity-title</i> argument defines the area addresses for the IS-IS area and the system ID of the router.</p> <p>Note For more details about the format of the <i>network-entity-title</i> argument, refer to the “Configuring ISO CLNS” chapter in the Release 12.2, <i>Cisco IOS Apollo Domain, Banyan VINES, DECnet, IOS CLNS, XNS Configuration Guide</i> document.</p>

Assigning an IPv6 IS-IS Routing Process to an Interface

The second task in configuring IPv6 IS-IS requires assigning the IPv6 IS-IS routing process to an interface. Other routing protocols usually assign the routing process to a network.

To configure IPv6 IS-IS on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface <i>type number</i>	Specifies the interface type and number, and enters interface configuration mode.
Step 2	Router(config-if)# ipv6 address <i>ipv6-prefix/prefix-length [eui-64]</i>	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface. Note See the “ Enabling IPv6 Routing and Configuring IPv6 Addressing ” section on page 2 for more information on configuring IPv6 addresses.
Step 3	Router(config-if)# ipv6 router isis <i>area-name</i>	Enables the specified IPv6 IS-IS routing process on an interface. The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.

Configuring Administrative Distance for IPv6 IS-IS

To configure a new administrative distance for IPv6 IS-IS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router isis <i>area-name</i>	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode. The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and enters address family configuration mode. The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# distance <i>value</i>	Defines an administrative distance for IPv6 IS-IS routes in the IPv6 routing table. The <i>value</i> argument is an integer from 10 to 254. (The values 0 to 9 are reserved for internal use).

Configuring the Maximum Number of Parallel Routes for IPv6 IS-IS

To configure the maximum numbers of parallel paths that IPv6 IS-IS will support, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router isis <i>area-name</i>	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode. The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and enters address family configuration mode. The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# maximum-paths <i>number-paths</i>	Defines the maximum number of parallel routes that IPv6 IS-IS can support. The <i>number-paths</i> argument is an integer from 1 to 4.

Configuring Summary Prefixes for IPv6 IS-IS

To configure summary prefixes for IPv6 IS-IS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router isis <i>area-name</i>	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode. The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.

	Command	Purpose
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and enters address family configuration mode. The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# summary-prefix <i>ipv6-prefix/prefix-length</i> [level-1 level-1-2 level-2]	Creates summary prefixes of IPv6 Level 1 interface prefixes and IPv6 prefixes learned from Level 1 link-state packets (LSPs) by Level 2 routers. The <i>ipv6-prefix</i> argument in the summary-prefix command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The <i>prefix-length</i> argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.

Disabling IPv6 Protocol-Support Consistency Checks

Under IPv6 address family configuration, protocol-support consistency checks on packets received from adjacent neighbors can be disabled.



Note

Disabling the **adjacency-check** command can adversely affect your network configuration. Issue the **no adjacency-check** command when you are running IPv4 IS-IS on all your routers, and you want to add IPv6 IS-IS to your network but you need to maintain all your adjacencies during the transition. When the IPv6 IS-IS configuration is complete, remove the **no adjacency-check** command from the configuration.

To disable protocol-support consistency checks, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router isis <i>area-name</i>	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode. The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.

	Command	Purpose
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and enters address family configuration mode. The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# no adjacency-check	Disables the IPv6 protocol-support consistency checks performed on hello packets, allowing IPv6 to be introduced into an IPv4-only network without disrupting existing adjacencies. The adjacency-check command is enabled by default.

Originating a Default IPv6 Route

To configure an IS-IS instance to advertise the default IPv6 route (::/0), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router isis <i>area-name</i>	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode. The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and enters address family configuration mode. The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# default-information originate [route-map <i>map-name</i>]	Injects a default IPv6 route into an IS-IS routing domain. The route-map keyword and <i>map-name</i> argument specify the conditions under which the IPv6 default route is advertised. If the route map keyword is omitted, then the IPv6 default route will be unconditionally advertised at Level 2.

Redistributing Routes into an IPv6 IS-IS Routing Process

To redistribute IPv6 routes between protocols, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router isis <i>area-name</i>	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode. The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and enters address family configuration mode. The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# redistribute <i>protocol</i> [<i>process-id</i>] { level-1 [into level-2] level-1-2 level-2 [into level-1]} [metric <i>metric-value</i>] [metric-type { internal external }] [route-map <i>map-name</i>]	Redistributes a specified route into the specified IPv6 IS-IS routing process. The <i>protocol</i> argument can be one of the following keywords: bgp , connected , isis , rip , or static . Note IPv6 IS-IS will ignore any configured redistribution of connected routes. IS-IS only advertises prefixes on an interface if IS-IS is running over the interface, or if the interface has been configured as passive.

Redistributing IPv6 IS-IS Routes Between IS-IS Levels

To redistribute IPv6 routes learned at one IS-IS level into a different level, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router isis <i>area-name</i>	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode. The <i>area-name</i> argument is required for multiarea IS-IS configuration. The name must be unique among all IP or CLNS router processes for a given router.

	Command	Purpose
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and enters address family configuration mode. The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# redistribute <i>protocol</i> { level-1 [into level-2] level-2 [into level-1]} distribute-list <i>prefix-list-name</i>	Redistributes IPv6 routes from one IS-IS level into another IS-IS level. By default, the routes learned by Level 1 instances are redistributed by the Level 2 instance. Note The <i>protocol</i> argument must be isis in this configuration of the redistribute command. Only the arguments and keywords relevant to this task are specified here.

Verifying IPv6 IS-IS

Enter the **show ipv6 protocols** privileged EXEC command to display the parameters and current state of the active IPv6 routing process:

```
Router# show ipv6 protocols

IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "isis"
  Interfaces:
    Ethernet0/0/3
    Ethernet0/0/1
    Serial1/0/1
    Loopback1 (Passive)
    Loopback2 (Passive)
    Loopback3 (Passive)
    Loopback4 (Passive)
    Loopback5 (Passive)
  Redistribution:
    Redistributing protocol static at level 1
  Address Summarization:
    L2: 33::/16 advertised with metric 0
    L2: 44::/16 advertised with metric 20
    L2: 66::/16 advertised with metric 10
    L2: 77::/16 advertised with metric 10
```

Enter the **show isis topology** privileged EXEC command to display a list of all connected routers running IS-IS in all areas:

```
Router# show isis topology

IS-IS paths to level-1 routers
System Id      Metric  Next-Hop      Interface      SNPA
0000.0000.000C
0000.0000.000D  20     0000.0000.00AA Sel/0/1        *HDLC*
0000.0000.000F  10     0000.0000.000F Et0/0/1        0050.e2e5.d01d
0000.0000.00AA  10     0000.0000.00AA Sel/0/1        *HDLC*
```

```
IS-IS paths to level-2 routers
System Id      Metric  Next-Hop      Interface      SNPA
0000.0000.000A 10      0000.0000.000A Et0/0/3        0010.f68d.f063
0000.0000.000B 20      0000.0000.000A Et0/0/3        0010.f68d.f063
0000.0000.000C --
0000.0000.000D 30      0000.0000.000A Et0/0/3        0010.f68d.f063
0000.0000.000E 30      0000.0000.000A Et0/0/3        0010.f68d.f063
```

Enter the **show clns is-neighbors** privileged EXEC command to confirm that the local router has formed all the necessary IS-IS adjacencies with other IS-IS neighbors. To display the IPv6 link-local addresses of the neighbors, specify the **detail** keyword.

```
Router# show clns is-neighbors detail

System Id      Interface  State  Type  Priority  Circuit Id      Format
0000.0000.00AA Se1/0/1    Up     L1    0         00              Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::D0:D37C:C854:5
  Uptime: 17:21:38
0000.0000.000F Et0/0/1    Up     L1    64      0000.0000.000C.02 Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::250:E2FF:FEE5:D01D
  Uptime: 17:21:41
0000.0000.000A Et0/0/3    Up     L2    64      0000.0000.000C.01 Phase V
  Area Address(es): 49.000b
  IPv6 Address(es): FE80::210:F6FF:FE8D:F063
  Uptime: 17:22:06
```

Enter the **show isis database detail** privileged EXEC command to display LSPs received from other routers and the IPv6 prefixes they are advertising:

```
Router# show isis database detail

IS-IS Level-1 Link State Database:
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0000.000C.00-00* 0x00000056   0x2445        901            1/0/0
  Area Address: 49.0001
  NLPID:        0x8E
  IPv6 Address: 33:1:3:3:3:3:3:3
  Metric: 10    IPv6 3003:6::/64
  Metric: 10    IPv6 3003:7::/64
  Metric: 0     IPv6 33:1:3:3:3:3:3:3/128
  Metric: 0     IPv6 33:2:3:3:3:3:3:3/128
  Metric: 0     IPv6 33:3:3:3:3:3:3:3/128
  Metric: 0     IPv6 33:4:3:3:3:3:3:3/128
  Metric: 0     IPv6 33:5:3:3:3:3:3:3/128
  Metric: 10    IS-Extended 0000.0000.000C.02
  Metric: 10    IS-Extended 0000.0000.000C.01
  Metric: 10    IS-Extended 0000.0000.00AA.00
0000.0000.000C.02-00* 0x00000051   0xE1EB        1026           0/0/0
  Metric: 0     IS-Extended 0000.0000.000C.00
  Metric: 0     IS-Extended 0000.0000.000F.00

IS-IS Level-2 Link State Database:
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0000.000A.00-00 0x00000059   0x378A        949            0/0/0
  Area Address: 49.000b
  NLPID:        0x8E
  IPv6 Address: 11:1:1:1:1:1:1:1
  Metric: 10    IPv6 2001:2::/64
  Metric: 10    IPv6 3001:3::/64
  Metric: 10    IPv6 3001:2::/64
  Metric: 10    IS-Extended 0000.0000.000A.01
  Metric: 10    IS-Extended 0000.0000.000B.00
  Metric: 10    IS-Extended 0000.0000.000C.01
```

```

Metric: 0          IPv6 11:1:1:1:1:1:1:1/128
Metric: 0          IPv6 11:2:1:1:1:1:1:1/128
Metric: 0          IPv6 11:3:1:1:1:1:1:1/128
Metric: 0          IPv6 11:4:1:1:1:1:1:1/128
Metric: 0          IPv6 11:5:1:1:1:1:1:1/128
0000.0000.000A.01-00 0x00000050 0xB0AF 491 0/0/0
Metric: 0          IS-Extended 0000.0000.000A.00
Metric: 0          IS-Extended 0000.0000.000B.00

```

IPv6 IS-IS Configuration Examples

This section provides the following IPv6 integrated IS-IS configuration examples:

- [IS-IS Interface Configuration Example](#)
- [IPv6 IS-IS Interface Configuration Example](#)
- [IPv6 IS-IS Administrative Distance Configuration Example](#)
- [IPv6 IS-IS Maximum Parallel Routes Configuration Example](#)
- [IPv6 IS-IS Summary Prefix Configuration Example](#)
- [Disabling IPv6 Protocol-Support Consistency Checks Configuration Example](#)
- [IPv6 IS-IS Default Route Origination Configuration Example](#)
- [Redistributing IPv6 Routes Example](#)
- [Redistributing IPv6 Routes Between IS-IS Levels Example](#)

IS-IS Interface Configuration Example

In the following example, IS-IS is configured with an area name and the NET is defined:

```

router isis
 net 49.0001.0000.0000.000c.00

```

IPv6 IS-IS Interface Configuration Example

In the following example, IPv6 is globally enabled, an IPv6 address is configured on an interface, and the interface is configured to run IPv6 IS-IS:

```

ipv6 unicast-routing
!
interface Ethernet0/0/1
 ipv6 address 3003:6::3/64
 ipv6 router isis

```

IPv6 IS-IS Administrative Distance Configuration Example

In the following example, the IPv6 IS-IS administrative distance is set to 90:

```

router isis
 address-family ipv6
 distance 90
 exit-address-family

```

IPv6 IS-IS Maximum Parallel Routes Configuration Example

In the following example, the maximum number of parallel paths is set to 3:

```
router isis
 address-family ipv6
 maximum-paths 3
 exit-address-family
```

IPv6 IS-IS Summary Prefix Configuration Example

In the following example, a prefix summary for Level 2 IPv6 IS-IS routes is set:

```
router isis
 address-family ipv6
 summary-prefix 3003:6::/24
 exit-address-family
```

Disabling IPv6 Protocol-Support Consistency Checks Configuration Example

In the following example, the **adjacency-check** command is disabled to allow a network administrator to configure IPv6 IS-IS on the router without disrupting the existing adjacencies:

```
router isis
 address-family ipv6
 no adjacency-check
 exit-address-family
```

IPv6 IS-IS Default Route Origination Configuration Example

In the following example, the IPv6 default route (::/0)—with an origin of Ethernet interface 0/0/1—is advertised with all other routes in router updates sent on Ethernet interface 0/0/1:

```
router isis
 address-family ipv6
 default-information originate
```

Redistributing IPv6 Routes Example

In the following example, IPv6 Border Gateway Protocol (BGP) routes are redistributed into the IPv6 IS-IS Level 2 routing process:

```
router isis
 address-family ipv6
 redistribute bgp 64500 metric 100 route-map isismap
 exit-address-family
```

Redistributing IPv6 Routes Between IS-IS Levels Example

In the following example, IPv6 IS-IS Level 1 routes are redistributed into the IPv6 IS-IS Level 2 routing process:

```
router isis
 address-family ipv6
 redistribute isis level-1 into level-2 distribute-list levelone
 exit-address-family
```

Configuring Multiprotocol BGP Extensions for IPv6

This section describes how to configure multiprotocol BGP extensions for IPv6. For a complete description of the multiprotocol BGP commands that appear in this section, refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com. For a complete description of the IPv4 BGP and multiprotocol BGP commands available in the Cisco IOS software, refer to the “BGP Commands” chapter and the “Multiprotocol BGP Extensions for IP Multicast Commands” chapter of the Release 12.2, *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*. To locate documentation for all other IPv4 commands, use the command reference master index or search online.

Multiprotocol BGP in IPv6 functions the same and offers the same benefits as multiprotocol BGP in IPv4. IPv6 enhancements to multiprotocol BGP include support for an IPv6 address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses and scoped addresses (the next hop attribute uses a global IPv6 addresses and potentially a link-local address when a peer is reachable on the local link).



Note

The following sections describe the configuration tasks for creating an IPv6 multiprotocol routing process and associating peers, peer groups, and networks to the routing process. The following sections do not provide in-depth information on customizing multiprotocol BGP because the protocol functions the same in IPv6 as it does in IPv4. Refer to the “Configuring Multiprotocol BGP Extensions for IP Multicast” chapter of the Release 12.2, *Cisco IOS IP Configuration Guide* publication for information on customizing BGP and multiprotocol BGP extensions. Wherever applicable, the “Usage Guidelines” section of each command reference page in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document lists the equivalent IPv4 multiprotocol BGP command for each new IPv6 multiprotocol BGP command. Refer to the “Multiprotocol BGP Extensions for IP Multicast Commands” chapter of the Release 12.2, *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols* publication as you customize your IPv6 multiprotocol BGP routing process. Additionally, refer to the “Configuring IP Routing Protocol-Independent Features” chapter of the Release 12.2, *Cisco IOS IP Configuration Guide* publication for protocol-independent features that also apply to multiprotocol BGP.

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

- [Configuring an IPv6 BGP Routing Process](#) (required)
- [Configuring an IPv6 Multiprotocol BGP Peer](#) (required)
- [Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address](#) (optional)
- [Configuring an IPv6 Multiprotocol BGP Peer Group](#) (optional)
- [Advertising Routes into IPv6 Multiprotocol BGP](#) (required)
- [Configuring Route Maps for IPv6 Multiprotocol BGP Prefixes](#) (optional)
- [Redistributing Prefixes into IPv6 Multiprotocol BGP](#) (optional)
- [Configuring a BGP Router ID](#) (optional)
- [Verifying IPv6 Multiprotocol BGP Configuration](#) (optional)

See the “[IPv6 Multiprotocol BGP Configuration Examples](#)” section for configuration examples.

Configuring an IPv6 BGP Routing Process

To configure an IPv6 BGP routing process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router bgp <i>autonomous-system</i>	Configures a BGP routing process, and places the router in router configuration mode for the specified routing process.
Step 2	Router(config-router)# no bgp default ipv4-unicast	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. Note Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the neighbor remote-as router configuration command unless you configure the no bgp default ipv4-unicast router configuration command before configuring the neighbor remote-as command.

Configuring an IPv6 Multiprotocol BGP Peer

To configure IPv6 multiprotocol BGP between two IPv6 routers (peers), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router bgp <i>autonomous-system</i>	Places the router in router configuration mode for the specified routing process.
Step 2	Router(config-router)# neighbor <i>ipv6-address remote-as</i> <i>autonomous-system-number</i>	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. The <i>ipv6-address</i> argument in the neighbor remote-as command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 3	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family and places the router in address family configuration mode. The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 4	Router(config-router-af)# neighbor <i>ipv6-address activate</i>	Enables the neighbor to exchange prefixes for the IPv6 address family with the local router.



Note

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address

Configuring IPv6 multiprotocol BGP between two IPv6 routers (peers) using link-local addresses requires that the interface for the neighbor be identified by using the **update-source** router configuration command and that a route map be configured to set an IPv6 global next hop.

To configure IPv6 multiprotocol BGP between two peers using link-local addresses, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router bgp <i>autonomous-system</i>	Places the router in router configuration mode for the specified routing process.
Step 2	Router(config-router)# neighbor <i>ipv6-address</i> remote-as <i>autonomous-system-number</i>	Adds the link-local IPv6 address of the neighbor in the specified remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. The <i>ipv6-address</i> argument in the neighbor remote-as command must be a link-local IPv6 address in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 3	Router(config-router)# neighbor <i>ipv6-address</i> update-source <i>interface-type</i>	Specifies the link-local address over which the peering is to occur. If there are multiple connections to the neighbor and you do not specify the neighbor interface by using the <i>interface-type</i> argument in the neighbor update-source command, a TCP connection cannot be established with the neighbor using link-local addresses.
Step 4	Router(config-router)# address-family <i>ipv6</i> [unicast]	Specifies the IPv6 address family, and places the router in address family configuration mode. The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 5	Router(config-router-af)# neighbor <i>ipv6-address</i> activate	Enables the neighbor to exchange prefixes for the IPv6 address family with the local router using the specified link-local addresses.
Step 6	Router(config-router-af)# neighbor <i>ipv6-address</i> route-map <i>route-map-name</i> { in out }	Applies a route map to incoming or outgoing routes.
Step 7	Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode. Repeat this step to exit router configuration mode and return the router to global configuration mode.
Step 8	Router(config)# route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]	Defines a route map and places the router in route-map configuration mode. Follow this step with a match command.

	Command	Purpose
Step 9	Router(config-route-map)# match ipv6 address prefix-list <i>prefix-list-name</i>	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets. Follow this step with a set command.
Step 10	Router(config-route-map)# set ipv6 address next-hop <i>ipv6-address</i> [<i>link-local-address</i>]	Indicates where to output IPv6 packets that pass a match clause of a route map for policy routing. The <i>ipv6-address</i> argument specifies the IPv6 global address of the next hop to which packets are output. It need not be an adjacent router. The <i>link-local-address</i> argument specifies the IPv6 link-local address of the next hop to which packets are output. It must be an adjacent router. Note The route map sets the IPv6 next hop addresses (global and link-local) in BGP updates. If the route map is not configured, the next hop address in the BGP updates defaults to the unspecified IPv6 address (::) and is ignored. If you specify only the global IPv6 next hop address (the <i>ipv6-address</i> argument) with the set ipv6 next-hop command after specifying the neighbor interface (the <i>interface-type</i> argument) with the neighbor update-source command in Step 3 , the link-local address of the interface specified with the <i>interface-type</i> argument is included as the next hop in the BGP updates. Therefore, only one route map that sets the global IPv6 next hop address in BGP updates is required for multiple BGP peers that use link-local addresses.

**Note**

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, you must activate neighbors using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

**Note**

By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.

Configuring an IPv6 Multiprotocol BGP Peer Group

To configure an IPv6 peer group to perform multiprotocol BGP routing, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router bgp <i>autonomous-system</i>	Places the router in router configuration mode for the specified BGP routing process.
Step 2	Router(config-router)# neighbor <i>peer-group-name peer-group</i>	Creates a multiprotocol BGP peer group.
Step 3	Router(config-router)# neighbor <i>ipv6-address</i> remote-as <i>autonomous-system-number</i>	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. The <i>ipv6-address</i> argument in the neighbor remote-as command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 4	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and places the router in address family configuration mode. The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 5	Router(config-router-af)# neighbor <i>peer-group-name activate</i>	Enables the peer group to exchange prefixes for the specified family type with the neighbor and the local router.
Step 6	Router(config-router-af)# neighbor <i>ipv6-address peer-group peer-group-name</i>	Assigns the IPv6 address of a BGP neighbor to a peer group.



Note

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, you must activate neighbors using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.



Note

By default, peer groups that are defined in router configuration mode using the **neighbor peer-group** command exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, you must activate peer groups using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

Members of a peer group automatically inherit the address prefix configuration of the peer group.

Refer to the section “Configure BGP Peer Groups” of the “Configuring BGP” chapter in the Release 12.2, *Cisco IOS IP Configuration Guide* for more information on assigning options to peer groups and making a BGP or multiprotocol BGP neighbor a member of a peer group.

Advertising Routes into IPv6 Multiprotocol BGP

To advertise (inject) a prefix into IPv6 multiprotocol BGP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router bgp <i>autonomous-system</i>	Places the router in router configuration mode for the specified BGP routing process.
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and places the router in address family configuration mode. The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# network <i>ipv6-prefix/prefix-length</i>	Advertises (injects) the specified prefix into the IPv6 BGP database. (The routes must first be found in the unicast routing table.) Specifically, the prefix is injected into the database for the address family specified in the previous step. Routes are tagged from the specified prefix as “local origin.” The <i>ipv6-prefix</i> argument in the network command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The <i>prefix-length</i> argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.



Note

By default, networks that are defined in router configuration mode using the **network** command are injected into the IPv4 unicast database. To inject a network into another database, such as the IPv6 BGP database, you must define the network using the **network** command in address family configuration mode for the other database, as shown for the IPv6 BGP database.

Configuring Route Maps for IPv6 Multiprotocol BGP Prefixes

To configure a route map for IPv6 multiprotocol BGP prefixes, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router bgp <i>autonomous-system</i>	Places the router in router configuration mode for the specified BGP routing process.
Step 2	Router(config-router)# neighbor <i>ipv6-address remote-as</i> <i>autonomous-system-number</i>	Adds the IPv6 address of the neighbor in the remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.

	Command	Purpose
Step 3	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and places the router in address family configuration mode. The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 4	Router(config-router-af)# neighbor <i>ipv6-address</i> activate	Enables the IPv6 address family for the neighbor in the remote autonomous system.
Step 5	Router(config-router-af)# neighbor <i>ipv6-address</i> route-map <i>route-map-name</i> { in out }	Applies a route map to incoming or outgoing routes.
Step 6	Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode. Repeat this step to exit router configuration mode and return the router to global configuration mode.
Step 7	Router(config)# route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]	Defines a route map, and places the router in route-map configuration mode. Follow this step with a match command.
Step 8	Router(config-route-map)# match ipv6 address <i>prefix-list</i> <i>prefix-list-name</i>	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets.

**Note**

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, you must activate neighbors using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

**Note**

By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.

Redistributing Prefixes into IPv6 Multiprotocol BGP

Redistribution is the process of injecting prefixes from one routing protocol into another routing protocol. The following steps explain how to inject prefixes from a routing protocol into IPv6 multiprotocol BGP. Specifically, prefixes that are redistributed into IPv6 multiprotocol BGP using the **redistribute** router configuration command are injected into the IPv6 unicast database.

To redistribute (inject) prefixes from a routing protocol into IPv6 multiprotocol BGP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router bgp <i>autonomous-system</i>	Places the router in router configuration mode for the specified BGP routing process.
Step 2	Router(config-router)# address-family ipv6 [unicast]	Specifies the IPv6 address family, and places the router in address family configuration mode. The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 3	Router(config-router-af)# redistribute <i>protocol</i> [<i>process-id</i>] { level-1 level-1-2 level-2 } [metric <i>metric-value</i>] [metric-type { internal external }] [route-map <i>map-name</i>]	Specifies the routing protocol from which prefixes should be redistributed into IPv6 multiprotocol BGP. The <i>protocol</i> argument can be one of the following keywords: bgp , connected , isis , rip , or static . Note The connected keyword refers to routes that are established automatically by IPv6 having been enabled on an interface.

Configuring a BGP Router ID

BGP uses a router ID to identify BGP-speaking peers. By default, the Cisco IOS software sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the router, then the software chooses the highest IPv4 address configured to a physical interface on the router. When configuring BGP on a router that is enabled only for IPv6 (IPv4 processing is not enabled on the router), you must manually configure an IPv4 address as the BGP router ID for the router. The IPv4 address that you configure as the BGP router ID for the router must be unique to the BGP peers of the router.

To configure a fixed IPv4 router ID for a BGP-speaking router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# router bgp <i>autonomous-system</i>	Places the router in router configuration mode for the specified BGP routing process.
Step 2	Router(config-router)# bgp router-id <i>ip-address</i>	Configures a fixed IPv4 router ID as the identifier of the local router running BGP. Note Configuring a router ID using the bgp router-id command resets all active BGP peering sessions.

Verifying IPv6 Multiprotocol BGP Configuration

To display entries in the IPv6 BGP routing table, enter the **show bgp ipv6** privileged EXEC command:

```
Router# show bgp ipv6
```

```
BGP table version is 12612, local router ID is 192.31.7.225
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*>
*                   3FFE:C00:E:C::2          0 3748 4697 1752 i
*                   3FFE:1100:0:CC00::1          0 1849 1273 1752 i
* 2001:618:3::/48   3FFE:C00:E:4::2          1 0 4554 1849 65002 i
*>
*                   3FFE:1100:0:CC00::1          0 1849 65002 i
* 2001:620::/35    3FFE:80A0:0:F004::1          0 3320 1275 559 i
*                   3FFE:C00:E:9::2          0 1251 1930 559 i
*                   3FFE:3600::A            0 3462 10566 1930 559 i
*                   3FFE:700:20:1::11          0 293 1275 559 i
*                   3FFE:C00:E:4::2          1 0 4554 1849 1273 559 i
*                   3FFE:C00:E:B::2          0 237 3748 1275 559 i
*                   3FFE:C00:E:C::2          0 3748 1275 559 i
```



Note

For a description of each output display field, refer to the **show bgp ipv6** command in the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com.

Enter the **show bgp ipv6 summary** privileged EXEC command to display the status of all IPv6 BGP connections:

```
Router# show bgp ipv6 summary
```

```
BGP router identifier 192.168.7.225, local AS number 109
BGP table version is 21898, main routing table version 21898
175 network entries and 693 paths using 66927 bytes of memory
555 BGP path attribute entries using 29224 bytes of memory
539 BGP AS-PATH entries using 13620 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 10 history paths, 31 dampened paths
BGP activity 2672/8496 prefixes, 21621/20928 paths, scan interval 15 secs

Neighbor          V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3FFE:700:20:1::11  4     4    293   31525   14358  21898  0 0 14:29:34    77
3FFE:C00:E:0:1::1  4     4   4768     0         0     0  0 0 never Active
FE80::202:4BFF:FE1A:E42
                   4    30   4442   4442     5     0     0 3d01h      3
FE80::204:6DFF:FE25:6000
                   4    20   4443   4444     5     0     0 3d01h      5
```

To display IPv6 BGP dampened routes, enter the **show bgp ipv6 dampened-paths** privileged EXEC command:

```
Router# show bgp ipv6 dampened-paths
```

```
BGP table version is 12610, local router ID is 192.31.7.225
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	From	Reuse	Path
*d 3FFE:1000::/24	3FFE:C00:E:B::2	00:00:10	237 2839 5609 i
*d 2001:228::/35	3FFE:C00:E:B::2	00:23:30	237 2839 5609 2713 i

To display debug messages for IPv6 BGP dampening packets, enter the **debug bgp ipv6 dampening** privileged EXEC command:



Note

By default, the network server sends the output from debug commands and system error messages to the console. To redirect debug output, use the **logging** command options within configuration mode. Possible destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server. For complete information on debug commands and redirecting debug output, refer to the Release 12.2, *Cisco IOS Debug Command Reference*.

```
Router# debug bgp ipv6 dampening
```

```
00:13:28:BGP(1):charge penalty for 2000:0:0:1::/64 path 2 1 with half-life-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:13:28:BGP(1):charge penalty for 2000:0:0:1:1::/80 path 2 1 with half-life-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:13:28:BGP(1):charge penalty for 2000:0:0:5::/64 path 2 1 with half-life-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:16:03:BGP(1):charge penalty for 2000:0:0:1::/64 path 2 1 with half-life-time 15
reuse/suppress 750/2000
00:16:03:BGP(1):flapped 2 times since 00:02:35. New penalty is 1892
00:18:28:BGP(1):suppress 2000:0:0:1:1::/80 path 2 1 for 00:27:30 (penalty 2671)
00:18:28:half-life-time 15, reuse/suppress 750/2000
00:18:28:BGP(1):suppress 2000:0:0:1::/64 path 2 1 for 00:27:20 (penalty 2664)
00:18:28:half-life-time 15, reuse/suppress 750/2000
```

To display debug messages for IPv6 BGP update packets, enter the **debug bgp ipv6 updates** privileged EXEC command:

```
Router# debug bgp ipv6 updates
```

```
14:04:17:BGP(1):2000:0:0:2::2 computing updates, afi 1, neighbor version 0, table version
1, starting at ::
14:04:17:BGP(1):2000:0:0:2::2 update run completed, afi 1, ran for 0ms, neighbor version
0, start version 1, throttled to 1
14:04:19:BGP(1):sourced route for 2000:0:0:2::1/64 path #0 changed (weight 32768)
14:04:19:BGP(1):2000:0:0:2::1/64 route sourced locally
14:04:19:BGP(1):2000:0:0:2:1::/80 route sourced locally
14:04:19:BGP(1):2000:0:0:3::2/64 route sourced locally
14:04:19:BGP(1):2000:0:0:4::2/64 route sourced locally
14:04:22:BGP(1):2000:0:0:2::2 computing updates, afi 1, neighbor version 1, table version
6, starting at ::
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (format) 2000:0:0:2::1/64, next 2000:0:0:2::1,
metric 0, path
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (format) 2000:0:0:2:1::/80, next 2000:0:0:2::1,
metric 0, path
```

```
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (prepend, chgflags:0x208) 2000:0:0:3::2/64, next
2000:0:0:2::1, metric 0, path
14:04:22:BGP(1):2000:0:0:2::2 send UPDATE (prepend, chgflags:0x208) 2000:0:0:4::2/64, next
2000:0:0:2::1, metric 0, path
```

IPv6 Multiprotocol BGP Configuration Examples

This section provides the following multiprotocol BGP configuration examples:

- [IPv6 Multiprotocol BGP Peer Example](#)
- [IPv6 Multiprotocol BGP Peer Using Link-Local Addresses Example](#)
- [IPv6 Multiprotocol BGP Peer Group Example](#)
- [IPv6 Multiprotocol BGP Network Advertisement Example](#)
- [IPv6 Multiprotocol BGP Route Map Example](#)
- [IPv6 Multiprotocol BGP Route Redistribution Example](#)

IPv6 Multiprotocol BGP Peer Example

The following example configures the IPv6 multiprotocol BGP peer 3FFE:1100:0:CC00::1.

```
router bgp 100
no bgp default ipv4-unicast
neighbor 3FFE:1100:0:CC00::1 remote-as 20

address-family ipv6 unicast
neighbor 3FFE:1100:0:CC00::1 activate
```

IPv6 Multiprotocol BGP Peer Using Link-Local Addresses Example

The following example configures the IPv6 multiprotocol BGP peer FE80::250:BFF:FE0E:A471 over Fast Ethernet interface 0 and sets the route map named nh6 to include the IPv6 next hop global address of Fast Ethernet interface 0 in BGP updates. The IPv6 next hop link-local address can be set by the nh6 route map (not shown in the following example) or from the interface specified by the **neighbor update-source** router configuration command (as shown in the following example).

```
router bgp 170
neighbor FE80::250:BFF:FE0E:A471 remote-as 150
neighbor FE80::250:BFF:FE0E:A471 update-source fastether 0

address-family ipv6
neighbor FE80::250:BFF:FE0E:A471 activate
neighbor FE80::250:BFF:FE0E:A471 route-map nh6 out

route-map nh6
set ipv6 next-hop 3ffe:506::1
```



Note

If you specify only the global IPv6 next hop address (the *ipv6-address* argument) with the **set ipv6 next-hop** command after specifying the neighbor interface (the *interface-type* argument) with the **neighbor update-source** command, the link-local address of the interface specified with the *interface-type* argument is included as the next hop in the BGP updates. Therefore, only one route map that sets the global IPv6 next hop address in BGP updates is required for multiple BGP peers that use link-local addresses.

IPv6 Multiprotocol BGP Peer Group Example

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```
router bgp 100
no bgp default ipv4-unicast
neighbor group1 peer-group
neighbor 3FFE:1100:0:CC00::1 remote-as 20

address-family ipv6 unicast
neighbor group1 peer-group
neighbor group1 activate
neighbor 3FFE:1100:0:CC00::1 peer-group group1
```

IPv6 Multiprotocol BGP Network Advertisement Example

The following example injects the IPv6 network 3FFE:1100::/24 into the IPv6 unicast database of the local router. (BGP checks that a route for the network exists in the IPv6 unicast database of the local router before advertising the network.)

```
router bgp 100
no bgp default ipv4-unicast

address-family ipv6 unicast
network 3FFE:1100::/24

ipv6 prefix-list cisco seq 20 permit ::/0 le 128
```

IPv6 Multiprotocol BGP Route Map Example

The following example configures the route map named rtp to permit IPv6 unicast routes from network 3FFE:1100::/24 if they match the prefix list named cisco:

```
router bgp 109
no bgp default ipv4-unicast
neighbor 3FFE:1100:0:CC00::1 remote-as 20

address-family ipv6 unicast
neighbor 3FFE:1100:0:CC00::1 activate
neighbor 3FFE:1100:0:CC00::1 route-map rtp in

ipv6 prefix-list cisco seq 10 deny 3ffe:1100::/24

route-map rtp permit 10
match ipv6 address prefix-list cisco
```

IPv6 Multiprotocol BGP Route Redistribution Example

The following example redistributes RIP routes into the IPv6 unicast database of the local router:

```
router bgp 109
no bgp default ipv4-unicast
address-family ipv6 unicast
redistribute rip
```

Configuring Dual Protocol Stacks and IPv6 Overlay Tunnels

To support the transition from IPv4-only networks to integrated IPv4- and IPv6-based networks, the Cisco IOS software supports both the IPv4 and IPv6 protocol stacks, and overlay tunneling techniques. Supporting both the IPv4 and IPv6 protocol stacks in the Cisco IOS software enables a Cisco networking device to send and receive data on both IPv4 and IPv6 networks. Supporting manually configured, automatic, and 6to4 tunneling techniques enables a Cisco networking device to encapsulate IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the Internet).

The tasks in the following sections explain how to configure both the IPv4 and IPv6 protocol stacks and IPv6 overlay tunnels on a Cisco networking device. Each task in the list is identified as either required or optional:

- [Configuring IPv4 and IPv6 Protocol Stacks](#) (required)
- [Configuring IPv6 Overlay Tunnels](#) (required)
- [Verifying Dual Protocol Stack and IPv6 Overlay Tunnel Configuration](#) (optional)

See the “[Dual Protocol Stack and IPv6 Overlay Tunnel Configuration Examples](#)” section for configuration examples.

Configuring IPv4 and IPv6 Protocol Stacks

When an interface in a Cisco networking device is configured with both an IPv4 and an IPv6 address, the interface forwards both IPv4 and IPv6 traffic—the interface can send and receive data on both IPv4 and IPv6 networks. To configure an interface in a Cisco networking device to support both the IPv4 and IPv6 protocol stacks, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 2	Router(config)# interface <i>interface-type number</i>	Specifies the interface type and number, and places the router in interface configuration mode.
Step 3	Router(config-if)# ipv6 address <i>ipv6-prefix/prefix-length [eui-64]</i>	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface. Note See the “ Enabling IPv6 Routing and Configuring IPv6 Addressing ” section for more information on configuring IPv6 addresses.
Step 4	Router(config-if)# ip address <i>ip-address mask [secondary]</i>	Specifies a primary or secondary IPv4 address for an interface. Note Refer to the “Configuring IP Addressing” chapter of the Release 12.2, <i>Cisco IOS IP Configuration Guide</i> for information on configuring IPv4 addresses.

Configuring IPv6 Overlay Tunnels

With manually configured IPv6 tunnels, an IPv6 address is configured on a tunnel interface and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.

With automatic IPv6 tunnels, the tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of IPv4-compatible IPv6 addresses. The host or router at each end of an automatic tunnel must support both the IPv4 and IPv6 protocol stacks. Refer to the “Larger Address Space” section of the *IPv6 for Cisco IOS Software, File 1 of 3: Overview* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for more information on IPv4-compatible IPv6 addresses.

With 6to4 tunnels, the tunnel destination is determined by the IPv4 address of the border router that is concatenated to the prefix 2002::/16 in the format 2002:IPv4 address of the border router::/48. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks.

Configuring a Manual IPv6 Tunnel

To configure a manual IPv6 overlay tunnel, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface tunnel <i>tunnel-number</i>	Specifies a tunnel interface and number, and places the router in interface configuration mode.
Step 2	Router(config-if)# ipv6 address <i>ipv6-prefix/prefix-length [eui-64]</i>	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface. Note See the “ Enabling IPv6 Routing and Configuring IPv6 Addressing ” section for more information on configuring IPv6 addresses.
Step 3	Router(config-if)# tunnel source { <i>ip-address</i> <i>interface-type</i> <i>interface-number</i> }	Specifies the source IPv4 address or the source interface type and number for the tunnel interface.
Step 4	Router(config-if)# tunnel destination <i>ip-address</i>	Specifies the destination IPv4 address or host name for the tunnel interface.
Step 5	Router(config-if)# tunnel mode ipv6ip	Specifies a manual IPv6 tunnel. Note The tunnel mode ipv6ip command specifies IPv6 as the passenger protocol and IPv4 as both the encapsulation and transport protocol for the manual IPv6 tunnel. Substituting the tunnel mode gre ip command for the tunnel mode ipv6ip specifies generic routing encapsulation (GRE) as the encapsulation protocol for the tunnel.

Configuring an Automatic IPv6 Tunnel

To configure an automatic IPv6 overlay tunnel, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface tunnel <i>tunnel-number</i>	Specifies a tunnel interface and number, and places the router in interface configuration mode.
Step 2	Router(config-if)# tunnel source <i>interface-type interface-number</i>	Specifies the source interface type and number for the tunnel interface. Note The interface type and number specified in the tunnel source command must be configured with an IPv4 address and an IPv6 address.
Step 3	Router(config-if)# tunnel mode ipv6ip auto-tunnel	Specifies an IPv6 automatic tunnel using an IPv4-compatible IPv6 address.

Configuring a 6to4 Tunnel

To configure a 6to4 overlay tunnel, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# interface tunnel <i>tunnel-number</i>	Specifies a tunnel interface and number, and places the router in interface configuration mode.
Step 2	Router(config-if)# ipv6 unnumbered <i>interface-type interface-number</i>	Enables the processing of IPv6 packets on the tunnel interface without assigning an explicit IPv6 address to the tunnel interface. The <i>interface-type</i> and <i>interface-number</i> arguments specify the source address (global IPv6 address) that the unnumbered interface uses in the IPv6 packets that it originates. The source address cannot be another unnumbered interface.
Step 3	Router(config-if)# tunnel source <i>interface-type interface-number</i>	Specifies the source interface type and number for the tunnel interface. Note The interface type and number specified in the tunnel source command should be the same interface type and number specified in the ipv6 unnumbered command. Note The interface type and number specified in the tunnel source command must be configured with an IPv4 address and an IPv6 address.
Step 4	Router(config-if)# tunnel mode ipv6ip 6to4	Specifies an IPv6 automatic tunnel using a 6to4 address.

	Command	Purpose
Step 5	Router(config-if)# exit	Exits interface configuration mode, and returns the router to global configuration mode.
Step 6	Router(config)# ipv6 route 2002::/16 tunnel tunnel-number	Configures a static route for the IPv6 prefix 2002::/16 to the specified tunnel interface. Note When configuring a 6to4 overlay tunnel, you must configure a static route for the IPv6 prefix 2002::/16 to the 6to4 tunnel interface. The tunnel number specified in the ipv6 route command should be the same tunnel number specified in the interface tunnel command.

Verifying Dual Protocol Stack and IPv6 Overlay Tunnel Configuration

To verify that both the IPv4 and IPv6 protocol stacks are configured correctly on specific interfaces, and that IPv6 overlay tunnels are configured correctly, enter the **show running-config** privileged EXEC command. In the following example of the **show running-config** command, Ethernet interface 0 and FDDI interface 0 are configured with an IPv6 address. Additionally, tunnel interface 0, tunnel interface 1, and tunnel interface 2 are configured as manual, automatic, and 6to4 tunnels, respectively. An IPv6 static route is also configured for network 2002::/16 to interface tunnel 2 (the 6to4 tunnel).

```
Router# show running-config

Building configuration...

Current configuration : 22324 bytes
!
hostname router1
!
no ip bootp server
ipv6 unicast-routing
!
interface Tunnel0
  ipv6 address 3ffe:b00:c18:1::3/127
  tunnel source fddi0
  tunnel destination 172.16.11.21
  tunnel mode ipv6ip
!
interface Tunnel1
  tunnel source Fddi0
  tunnel mode ipv6ip auto-tunnel
!
interface Tunnel2
  ipv6 unnumbered Ethernet0
  tunnel source Ethernet0
  tunnel mode ipv6ip 6to4
!
interface Ethernet0
  ip address 192.168.99.1 255.255.255.0
  no ip route-cache
  no ip mroute-cache
  no keepalive
  media-type 10BaseT
  ipv6 enable
  ipv6 address 2002:c0a8:6301:1::/64 eui-64
```

```

!
interface Fddi0
 ip address 172.31.7.104 255.255.255.224
 no ip route-cache
 no ip mroute-cache
 no keepalive
 ipv6 address 3FFE:C00:0:2::/64 eui-64
!
ipv6 route 2002::/16 Tunnel2
!

```



Note Display text was omitted from the example.

Dual Protocol Stack and IPv6 Overlay Tunnel Configuration Examples

This section provides the following dual protocol stack and IPv6 overlay tunnel configuration examples:

- [Dual Protocol Stack Configuration Example](#)
- [Manually Configured IPv6 Tunnel Configuration Example](#)
- [Automatic IPv6 Tunnel Configuration Example](#)
- [6to4 Tunnel Configuration Example](#)

Dual Protocol Stack Configuration Example

The following example enables the forwarding of IPv6 unicast datagrams globally on the router and configures Ethernet interface 0 with both an IPv4 address and an IPv6 address:

```

ipv6 unicast-routing

interface Ethernet0
 ip address 192.168.99.1 255.255.255.0
 ipv6 address 3ffe:b00:c18:1::3/64

```

Manually Configured IPv6 Tunnel Configuration Example

The following example configures a manual IPv6 tunnel between router A and router B. In the example, tunnel interface 0 for both router A and router B is manually configured with a global IPv6 address. The tunnel source and destination addresses are also manually configured.

Router A Configuration

```

interface ethernet 0
 ip address 192.168.99.1 255.255.255.0

interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::3/127
 tunnel source ethernet 0
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip

```

Router B Configuration

```
interface ethernet 0
 ip address 192.168.30.1 255.255.255.0

interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::2/127
 tunnel source ethernet 0
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

Automatic IPv6 Tunnel Configuration Example

The following example configures an automatic IPv6 tunnel that uses Ethernet interface 0 as the tunnel source. The tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of an IPv4-compatible IPv6 address. Specifically, the IPv6 prefix 0:0:0:0:0 is concatenated to an IPv4 address (in the format 0:0:0:0:0:A.B.C.D or ::A.B.C.D) to create the IPv4-compatible IPv6 address. Ethernet interface 0 is configured with a global IPv6 address and an IPv4 address (the interface supports both the IPv6 and IPv4 protocol stacks). Multiprotocol BGP is used in the example to exchange IPv6 reachability information with the peer 10.67.0.2 over Ethernet interface 0. The IPv4 address of Ethernet interface 0 is used in the low-order 32 bits of an IPv4-compatible IPv6 address and is also used as the next hop attribute. Because an IPv4-compatible IPv6 address is used as the next hop attribute, BGP routing information can be used to automatically determine the IPv4 endpoint for the 6to4 tunnel.

```
interface tunnel 0
 tunnel source Ethernet 0
 tunnel mode ipv6ip auto-tunnel

interface ethernet 0
 ip address 10.27.0.1 255.255.255.0
 ipv6 address 3000::1/64

router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 neighbor ::10.67.0.2 remote-as 100

address-family ipv6
 neighbor ::10.67.0.2 activate
 neighbor ::10.67.0.2 next-hop-self
 network 2001:c001:d00d:b10b::/64
```

6to4 Tunnel Configuration Example

The following example configures a 6to4 tunnel between router A and router B. In the example, Ethernet interface 0 for both router A and router B is configured with a global IPv6 address from a top-level Internet service provider (ISP) and an IPv4 address. Tunnel interface 0 for both router A and router B is configured without an IPv4 or IPv6 address because the IPv4 or IPv6 addresses on Ethernet interface 0 of both routers are used to construct a tunnel source address. A tunnel destination address is not specified on either router because the destination address is automatically constructed. An IPv6 static route for network 2002::/16 to tunnel interface 0 is configured on both routers.

Router A Configuration

```
interface Ethernet 0
 ip address 192.168.99.1 255.255.255.0
 ipv6 address 2002:c0a8:6301:1::/64 eui-64

interface Tunnel 0
 ipv6 unnumbered Ethernet 0
 tunnel source Ethernet 0
 tunnel mode ipv6ip 6to4

ipv6 route 2002::/16 Tunnel 0
```

Router B Configuration

```
interface Ethernet 0
 ip address 192.168.30.1 255.255.255.0
 ipv6 address 2002:c0a8:1e01:1::/64 eui-64

interface Tunnel 0
 ipv6 unnumbered Ethernet 0
 tunnel source Ethernet 0
 tunnel mode ipv6ip 6to4

ipv6 route 2002::/16 Tunnel 0
```

Monitoring and Maintaining IPv6

To monitor and maintain IPv6, use the following commands in privileged EXEC mode, as needed:

**Note**

Refer to the *IPv6 for Cisco IOS Software, File 3 of 3: Commands* document in either the New Features in Release 12.2 T or New Features in Release 12.0 ST area of Cisco.com for the complete syntax of each command.

Command	Purpose
Router# clear bgp ipv6	Resets an IPv6 BGP connection by dropping all neighbor sessions.
Router# clear bgp ipv6 dampening	Clears IPv6 BGP route dampening information and unsuppresses the suppressed routes.
Router# clear bgp ipv6 external	Clears all external IPv6 BGP peers.
Router# clear bgp ipv6 flap-statistics	Clears IPv6 BGP flap statistics.
Router# clear ipv6 neighbors	Deletes all entries in the IPv6 neighbor discovery cache, except static entries
Router# clear ipv6 prefix-list	Resets the hit count of IPv6 prefix list entries.
Router# clear ipv6 route	Deletes routes from the IPv6 routing table.
Router# clear ipv6 traffic	Resets IPv6 traffic counters.
Router# debug bgp ipv6 dampening	Displays debug messages for IPv6 BGP dampening packets.
Router# debug bgp ipv6 updates	Displays debug messages for IPv6 BGP update packets.
Router# debug ipv6 icmp	Displays debug messages for IPv6 ICMP transactions.
Router# debug ipv6 nd	Displays debug messages for IPv6 ICMP neighbor discovery transactions.

Command	Purpose
Router# debug ipv6 packet	Displays debug messages for IPv6 packets.
Router# debug ipv6 rip	Displays debug messages for IPv6 RIP routing transactions.
Router# debug ipv6 routing	Displays debug messages for IPv6 routing table updates and route cache updates.
Router# show bgp ipv6	Displays entries in the IPv6 BGP routing table.
Router# show bgp ipv6 community	Displays routes that belong to specified IPv6 BGP communities.
Router# show bgp ipv6 community-list	Displays routes that are permitted by the IPv6 BGP community list.
Router# show bgp ipv6 dampened-paths	Displays IPv6 BGP dampened routes.
Router# show bgp ipv6 filter-list	Displays routes that conform to a specified IPv6 filter list.
Router# show bgp ipv6 flap-statistics	Displays IPv6 BGP flap statistics.
Router# show bgp ipv6 inconsistent-as	Displays IPv6 BGP routes with inconsistent originating autonomous systems.
Router# show bgp ipv6 neighbors	Displays information about IPv6 BGP connections to neighbors.
Router# show bgp ipv6 paths	Displays all the IPv6 BGP paths in the database.
Router# show bgp ipv6 quote-regexp	Displays IPv6 BGP routes matching the autonomous system-path regular expression as a quoted string of characters.
Router# show bgp ipv6 regexp	Displays IPv6 BGP routes matching the autonomous system-path regular expression.
Router# show bgp ipv6 summary	Displays the status of all IPv6 BGP connections.
Router# show cdp entry	Displays CDP entry information.
Router# show cdp neighbors detail	Displays detailed CDP neighbor information.
Router# show ip sockets	Displays IP socket information.
Router# show ipv6 access-list	Displays the contents of all current IPv6 access lists.
Router# show ipv6 cef	Displays FIB entries based on IPv6 address information.
Router# show ipv6 interface	Displays the usability status of interfaces configured for IPv6.
Router# show ipv6 mtu	Displays maximum transmission unit (MTU) cache information for IPv6 interfaces.
Router# show ipv6 neighbors	Displays IPv6 neighbor discovery cache information.
Router# show ipv6 prefix-list	Displays information about an IPv6 prefix list or IPv6 prefix list entries.
Router# show ipv6 protocols	Displays the parameters and current state of the active IPv6 routing protocol process.
Router# show ipv6 rip	Displays information about current IPv6 RIP processes.
Router# show ipv6 route	Displays the current contents of the IPv6 routing table.
Router# show ipv6 route summary	Displays the current contents of the IPv6 routing table in summary format.
Router# show ipv6 routers	Displays IPv6 router advertisement information received from onlink routers.
Router# show ipv6 traffic	Displays statistics about IPv6 traffic.
Router# show ipv6 tunnel	Displays IPv6 tunnel information.