



IP Routing Configuration Guide, Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches

First Published: 2021-08-19

Last Modified: 2023-06-02

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020–2023 Cisco Systems, Inc. All rights reserved.



CHAPTER 1

Configuring Bidirectional Forwarding Detection

- [Bidirectional Forwarding Detection, on page 1](#)

Bidirectional Forwarding Detection

This document describes how to enable the Bidirectional Forwarding Detection (BFD) protocol. BFD is a detection protocol that is designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols.

BFD provides a consistent failure detection method for network administrators, in addition to fast forwarding path failure detection. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and reconvergence time will be consistent and predictable.

Prerequisites for Bidirectional Forwarding Detection

- All participating switches must enable Cisco Express Forwarding and IP routing.
- Before BFD is deployed on a switch, it is necessary to configure one of the IP routing protocols that are supported by BFD. You should implement fast convergence for the routing protocol that you are using. See IP routing documentation for your version of Cisco IOS software for information on configuring fast convergence. See the "Restrictions for Bidirectional Forwarding Detection" section for more information on BFD routing protocol support in Cisco IOS software.

Restrictions for Bidirectional Forwarding Detection

- BFD support is not available for all platforms and interfaces. To confirm if a specific platform or interface has BFD support and to obtain the most accurate platform and hardware restrictions, see the Cisco IOS software release notes for your software version.
- BFD HA is not supported.
- BFD Echo Sessions scale: Up to 28 BFD Sessions of 100ms interval, Echo-mode BFD sessions allowed per device.
- As of IOS XE 17.5.1, only OSPF and OSPFv3 support for BFD are supported.
- Supported Time Intervals by port type:

Port Type	Minimum Interval
Routed-Ports	100ms
SVIs	100ms
L3 Port	250ms

Information About Bidirectional Forwarding Detection

The following sections provide information about bidirectional forwarding detection.

BFD Operation

BFD provides a low-overhead, short-duration method of detecting failures in the forwarding path between two adjacent devices. These devices include the interfaces, data links, and forwarding planes.

BFD is a detection protocol that you enable at the interface and routing protocol levels. Cisco supports BFD asynchronous mode. BFD asynchronous mode depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between devices. Therefore, in order to create a BFD session, you must configure BFD on both systems (or BFD peers). A BFD session is created once BFD is enabled on the interfaces and at the device level for the appropriate routing protocols. BFD timers are negotiated, and the BFD peers begin to send BFD control packets to each other at the negotiated interval.

Neighbor Relationships

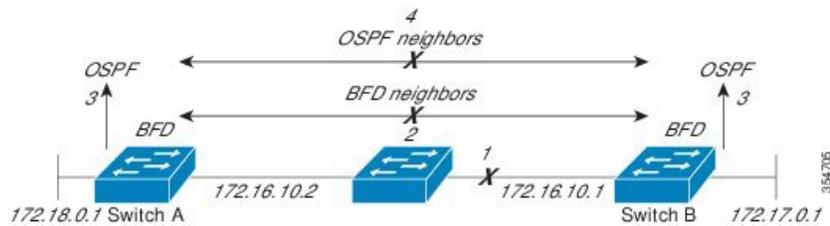
BFD provides fast BFD peer failure detection times independently. This is independent of all media types, encapsulations, topologies, and routing protocols such as BGP, EIGRP, IS-IS, and OSPF. BFD sends rapid failure detection notices to the routing protocols in the local device to initiate the routing table recalculation process. In this way, BFD contributes to greatly reduced overall network convergence time. The figure below shows a simple network with two devices running OSPF and BFD. When OSPF discovers a neighbor (1), it sends a request to the local BFD process. It initiates a BFD neighbor session with the OSPF neighbor device (2). The BFD neighbor session with the OSPF neighbor device is established (3).

Figure 1: BFD Process on a Network Configured with OSPF



The figure below shows what happens when a failure occurs in the network (1). The BFD neighbor session with the OSPF neighbor device is torn down (2). BFD notifies the local OSPF process that the BFD neighbor is no longer reachable (3). The local OSPF process tears down the OSPF neighbor relationship (4). If an alternative path is available, the devices immediately start converging on it.

Figure 2: BFD Process During a Network Failure



A routing protocol must register with BFD for every neighbor it acquires. Once a neighbor is registered, BFD initiates a session with the neighbor if a session does not already exist.

OSPF registers with BFD when:

- A neighbor finite state machine (FSM) transitions to full state.
- Both OSPF BFD and BFD are enabled.

On broadcast interfaces, OSPF establishes a BFD session only with the designated router (DR) and backup designated router (BDR). The session is not established between any two devices in a DROTHER state.

BFD Detection of Failures

Once a BFD session is established and timer negotiations are complete, BFD peers send BFD control packets. The packets act in the same manner as an IGP hello protocol to detect liveness, except at a more accelerated rate. The following information should be noted:

- BFD is a forwarding path failure detection protocol. BFD detects a failure, but the routing protocol must act to bypass a failed peer.
- Starting with Cisco IOS XE Denali 16.3.1, Cisco devices support BFD version 0. Devices use one BFD session for multiple client protocols in the implementation. For example, if a network is running OSPF and EIGRP across the same link to the same peer, only one BFD session is established. BFD shares session information with both routing protocols.

BFD Version Interoperability

All BFD sessions come up as Version 1 by default and are interoperable with Version 0. The system automatically performs BFD version detection, and BFD sessions between neighbors run in the highest common BFD version between neighbors. For example, if one BFD neighbor is running BFD Version 0 and the other BFD neighbor is running Version 1, the session runs BFD Version 0. The output from the **show bfd neighbors [details]** command verifies which BFD version a BFD neighbor is running.

See the "Example Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default" for an example of BFD version detection.

BFD Support for Nonbroadcast Media Interfaces

The BFD feature is supported on routed, SVI and L3 portchannels. The **bfd interval** command must be configured on the interface to initiate BFD monitoring.

How to Configure Bidirectional Forwarding Detection

The following sections provide configurational information about bidirectional forwarding detection.

Configuring BFD Session Parameters on the Interface

To configure BFD on an interface, you must set the baseline BFD session parameters. Repeat the steps in this procedure for each interface over which you want to run BFD sessions to BFD neighbors.

The following procedure shows BFD configuration steps for a physical interface. Please use the corresponding BFD timer values for SVIs and ether-channels respectively.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	Perform one of the following steps: <ul style="list-style-type: none"> • ip address <i>ipv4-address mask</i> • ipv6 address <i>ipv6-address/mask</i> Example: Configuring an IPv4 address for the interface: Device(config-if)# ip address 10.201.201.1 255.255.255.0 Configuring an IPv6 address for the interface: Device(config-if)# ipv6 address 2001:db8:1:1::1/32	Configures an IP address for the interface.
Step 4	bfd interval <i>milliseconds min_rx milliseconds multiplier interval-multiplier</i> Example: Device(config-if)# bfd interval 100 min_rx 100 multiplier 3	Enables BFD on the interface. The BFD interval configuration is removed when the subinterface on which it is configured is removed. The BFD interval configuration is not removed when: <ul style="list-style-type: none"> • An interface removes an IPv4 address. • An interface removes an IPv6 address is removed from an interface. • An interface disables IPv6. • An interface is shutdown • An interface globally or locally disables IPv4 CEF. • An interface globally or locally disables IPv6 CEF.

	Command or Action	Purpose
Step 5	end Example: Device(config-if) # end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring BFD Support for Dynamic Routing Protocols

Configuring BFD Support for OSPF

This section describes the procedures for configuring BFD support for OSPF so that OSPF is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. You can either configure BFD support for OSPF globally on all interfaces or configure it selectively on one or more interfaces.

There are two methods for enabling BFD support for OSPF:

- You can enable BFD for all of the interfaces for which OSPF is routing by using the **bfd all-interfaces** command in router configuration mode. You can disable BFD support on individual interfaces using the **ip ospf bfd [disable]** command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which OSPF is routing by using the **ip ospf bfd** command in interface configuration mode.

See the following sections for tasks for configuring BFD support for OSPF:

Configuring BFD Support for OSPF for All Interfaces

To configure BFD for all OSPF interfaces, perform the steps in this section.

If you do not want to configure BFD on all OSPF interfaces and would rather configure BFD support specifically for one or more interfaces, see the "Configuring BFD Support for OSPF for One or More Interfaces" section.

Before you begin

- OSPF must be running on all participating routers.
- The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the "Configuring BFD Session Parameters on the Interface" section for more information.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device>enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device#configure terminal	
Step 3	router ospf <i>process-id</i> Example: Device(config)#router ospf 4	Specifies an OSPF process and enters router configuration mode.
Step 4	bfd all-interfaces Example: Device(config-router)#bfd all-interfaces	Enables BFD globally on all interfaces associated with the OSPF routing process.
Step 5	exit Example: Device(config-router)#exit	(Optional) Returns the device to global configuration mode. Enter this command only if you want to perform Step 7 to disable BFD for one or more interfaces.
Step 6	interface <i>type number</i> Example: Device(config)#interface fastethernet 6/0	(Optional) Enters interface configuration mode. Enter this command only if you want to perform Step 7 to disable BFD for one or more interfaces.
Step 7	ip ospf bfd [disable] Example: Device(config-if)#ip ospf bfd disable	(Optional) Disables BFD on a per-interface basis for one or more interfaces associated with the OSPF routing process. Note You should use the disable keyword only if you enabled BFD on all of the interfaces that OSPF is associated with using the bfd all-interfaces command in router configuration mode.
Step 8	end Example: Device(config-if)#end	Exits interface configuration mode and returns the router to privileged EXEC mode.
Step 9	show bfd neighbors [details] Example: Device#show bfd neighbors detail	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 10	show ip ospf Example: Device#show ip ospf	(Optional) Displays information that can help verify if BFD for OSPF has been enabled.

Configuring OSPF Support for BFD over IPv4 for One or More Interfaces

To configure BFD on one or more OSPF interfaces, perform the steps in this section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf bfd** [**disable**]
5. **end**
6. **show bfd neighbors** [**details**]
7. **show ip ospf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device#configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)#interface fastethernet 6/0	Enters interface configuration mode.
Step 4	ip ospf bfd [disable] Example: Device(config-if)#ip ospf bfd	Enables or disables BFD on a per-interface basis for one or more interfaces associated with the OSPF routing process. Note Use the disable keyword only if you enable BFD on all of the interfaces that OSPF is associated with using the bfd all-interfaces command in router configuration mode.
Step 5	end Example: Device(config-if)#end	Exits interface configuration mode and returns the device to privileged EXEC mode.
Step 6	show bfd neighbors [details] Example: Device#show bfd neighbors details	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered.

	Command or Action	Purpose
		<p>Note If hardware-offloaded BFD sessions are configured with Tx and Rx intervals that are not multiples of 50 ms, the hardware intervals are changed. However, output from the show bfd neighbors details command displays only the configured intervals, not the interval values that change.</p>
Step 7	<p>show ip ospf</p> <p>Example:</p> <pre>Device#show ip ospf</pre>	(Optional) Displays information that can help verify if BFD support for OSPF has been enabled.

Configuring BFD Support for Static Routing

Perform this task to configure BFD support for static routing. Repeat the steps in this procedure on each BFD neighbor. For more information, see the "Example: Configuring BFD Support for Static Routing" section.

Procedure

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Device>enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<p>configure terminal</p> <p>Example:</p> <pre>Device#configure terminal</pre>	Enters global configuration mode.
Step 3	<p>interface <i>type number</i></p> <p>Example:</p> <pre>Device(config)#interface serial 2/0</pre>	Configures an interface and enters interface configuration mode.
Step 4	<p>Perform one of the following steps:</p> <ul style="list-style-type: none"> • ip address <i>ipv4-address mask</i> • ipv6 address <i>ipv6-address/mask</i> <p>Example:</p> <p>Configuring an IPv4 address for the interface:</p> <pre>Device(config-if)#ip address 10.201.201.1 255.255.255.0</pre> <p>Configuring an IPv6 address for the interface:</p>	Configures an IP address for the interface.

	Command or Action	Purpose
	Device(config-if)#ipv6 address 2001:db8:1:1::1/32	
Step 5	<p>bfd interval <i>milliseconds</i> min_rx <i>milliseconds</i> multiplier <i>interval-multiplier</i></p> <p>Example:</p> <pre>Device(config-if)#bfd interval 500 min_rx 500 multiplier 5</pre>	<p>Enables BFD on the interface.</p> <p>The bfd interval configuration is removed when the subinterface on which it is configured is removed.</p> <p>The bfd interval configuration is not removed when:</p> <ul style="list-style-type: none"> • an IPv4 address is removed from an interface • an IPv6 address is removed from an interface • IPv6 is disabled from an interface • an interface is shutdown • IPv4 CEF is disabled globally or locally on an interface • IPv6 CEF is disabled globally or locally on an interface
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-if)#exit</pre>	Exits interface configuration mode and returns to global configuration mode.
Step 7	<p>ip route static bfd <i>interface-type interface-number</i> <i>ip-address</i> [group <i>group-name</i> [passive]]</p> <p>Example:</p> <pre>Device(config)#ip route static bfd TenGigabitEthernet1/0/1 10.10.10.2 group group1 passive</pre>	<p>Specifies a static route BFD neighbor.</p> <ul style="list-style-type: none"> • The <i>interface-type</i>, <i>interface-number</i>, and <i>ip-address</i> arguments are required because BFD support exists only for directly connected neighbors.
Step 8	<p>ip route [vrf <i>vrf-name</i>] <i>prefix mask</i> {<i>ip-address</i> <i>interface-type interface-number [ip-address]</i>} [dhcp] [<i>distance</i>] [name <i>next-hop-name</i>] [permanent track <i>number</i>] [tag <i>tag</i>]</p> <p>Example:</p> <pre>Device(config)#ip route 10.0.0.0 255.0.0.0</pre>	Specifies a static route BFD neighbor.
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config)#exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 10	show ip static route Example: Device#show ip static route	(Optional) Displays static route database information.
Step 11	show ip static route bfd Example: Device#show ip static route bfd	(Optional) Displays information about the static BFD configuration from the configured BFD groups and nongroup entries.
Step 12	exit Example: Device#exit	Exits privileged EXEC mode and returns to user EXEC mode.

Configuring BFD Echo Mode

BFD echo mode is enabled by default, but you can disable it such that it can run independently in each direction.

BFD echo mode works with asynchronous BFD. Echo packets are sent by the forwarding engine and forwarded back along the same path in order to perform detection--the BFD session at the other end does not participate in the actual forwarding of the echo packets. The echo function and the forwarding engine are responsible for the detection process; therefore, the number of BFD control packets that are sent out between two BFD neighbors is reduced. In addition, because the forwarding engine is testing the forwarding path on the remote (neighbor) system without involving the remote system, there is an opportunity to improve the interpacket delay variance, thereby achieving quicker failure detection times than when using BFD Version 0 with BFD control packets for the BFD session.

Echo mode is described as without asymmetry when it is running on both sides (both BFD neighbors are running echo mode).

Prerequisites

- BFD must be running on all participating devices.
- Before using BFD echo mode, you must disable the sending of Internet Control Message Protocol (ICMP) redirect messages by entering the **no ip redirects** command, in order to avoid high CPU utilization.
- The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the Configuring BFD Session Parameters on the Interface section for more information.

Disabling BFD Echo Mode Without Asymmetry

The steps in this procedure show how to disable BFD echo mode without asymmetry—no echo packets will be sent by the device, and the device will not forward BFD echo packets that are received from any neighbor devices.

Repeat the steps in this procedure for each BFD Device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device>enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device#configure terminal	Enters global configuration mode.
Step 3	no bfd echo Example: Device(config)#no bfd echo	Disables BFD echo mode. • Use the no form to disable BFD echo mode.
Step 4	end Example: Device(config)#end	Exits global configuration mode and returns to privileged EXEC mode.

Creating and Configuring BFD Templates

You can configure a single-hop template to specify a set of BFD interval values. BFD interval values specified as part of the BFD template are not specific to a single interface.



Note Configuring bfd-template will disable echo mode.

Configuring a Single-Hop Template

Perform this task to create a BFD single-hop template and configure BFD interval timers.

Procedure

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

	Command or Action	Purpose
Step 3	bfd-template single-hop <i>template-name</i> Example: <pre>Device(config)#bfd-template single-hop bfdtemplate1</pre>	Creates a single-hop BFD template and enters BFD configuration mode.
Step 4	interval min-tx <i>milliseconds</i> min-rx <i>milliseconds</i> multiplier <i>multiplier-value</i> Example: <pre>Device(bfd-config)#interval min-tx 120 min-rx 100 multiplier 3</pre>	Configures the transmit and receive intervals between BFD packets, and specifies the number of consecutive BFD control packets that must be missed before BFD declares that a peer is unavailable.
Step 5	end Example: <pre>Device(bfd-config)#end</pre>	Exits BFD configuration mode and returns the device to privileged EXEC mode.

Monitoring and Troubleshooting BFD

This section describes how to retrieve BFD information for maintenance and troubleshooting. The commands in these tasks can be entered as needed, in any order desired.

This section contains information for monitoring and troubleshooting BFD for the following Cisco platforms:

Monitoring and Troubleshooting BFD

To monitor or troubleshoot BFD, perform one or more of the steps in this section.

SUMMARY STEPS

1. **enable**
2. **show bfd neighbors [details]**
3. **debug bfd [packet | event]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Device>enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show bfd neighbors [details] Example: <pre>Device#show bfd neighbors details</pre>	(Optional) Displays the BFD adjacency database. <ul style="list-style-type: none"> • The details keyword shows all BFD protocol parameters and timers per neighbor.

	Command or Action	Purpose
Step 3	debug bfd [packet event] Example: Device#debug bfd packet	(Optional) Displays debugging information about BFD packets.

Configuration Examples for Bidirectional Forwarding Detection

This section provides configuration examples for Bidirectional Forwarding Detection:



CHAPTER 2

Configuring IPv4 Policy-Based Routing

- [Information About Policy-Based Routing, on page 15](#)
- [Restrictions and Limitations for Policy-Based Routing, on page 16](#)
- [How to Configure PBR, on page 19](#)

Information About Policy-Based Routing



Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

PBR (Policy Based Routing) is a technique used to make routing decisions based on configured policies.

When a router or switch receives a packet, a forwarding decision is based on the destination IP address of the packet, which is used to look up an entry in a routing table. However, in some cases, there may be a need to forward the packet based on other criteria, for example, the source IP address and not the destination IP address. This permits routing of packets originating from different sources to different networks, even when the destinations are the same, and can be useful when interconnecting several private networks.

With PBR, you classify traffic using access control lists (ACLs) and then make traffic go through a different path. PBR is applied to incoming packets. All packets received on an interface with PBR enabled are passed through route maps. Based on the criteria defined in the route maps, packets are forwarded (routed) to the appropriate next hop.

- Route map statement marked as permit is processed as follows:
 - A match command can match on multiple ACLs. A route map statement can contain multiple match commands. Logical or algorithm function is performed across all the match commands to reach a permit or deny decision.

For example:

```
match ip address acl1 acl2
```

```
match ip address acl3
```



Note IPv6 is not supported.

A packet is permitted if it is permitted by `acl1` or `acl2` or `acl3`.

- If the decision reached is permit, then the action specified by the `set` command is applied on the packet.
- If the decision reached is deny, then the PBR action (specified in the `set` command) is not applied. Instead the processing logic moves forward to look at the next route-map statement in the sequence (the statement with the next higher sequence number). If no next statement exists, PBR processing terminates, and the packet is routed using the default IP routing table.

- For PBR, route-map statements and ACLs marked as deny are not supported.

You can use standard IP ACLs to specify match criteria for a source address or extended IP ACLs to specify match criteria based on an end station. The process proceeds through the route map until a match is found. If no match is found, normal destination-based routing occurs. There is an implicit deny at the end of the list of match statements.

If match clauses are satisfied, you can use a `set` clause to specify the IP addresses identifying the next hop router in the path. You can also set an IP precedence value using the precedence number or name.

Restrictions and Limitations for Policy-Based Routing

- To use PBR, you must have the Network Advantage license enabled on the switch.
- By default, Policy-Based Routing (PBR) is DISABLED on the switch. PBR is enabled when a route-map is configured and applied on interface.
- Packets that are generated by the switch (CPU), or local packets, are not normally policy-routed. When you globally enable local PBR on the switch, all unicast packets that originate on the switch are subject to local PBR. The protocols that are supported for local PBR are NTP, DNS, MSDP, SYSLOG and TFTP. Local PBR is disabled by default.
- The switch does not support **route-map deny** statements for PBR.
- Match ACLs with deny ACEs not supported.
- When a policy route map is applied to a physical interface, that interface cannot become a member of an EtherChannel.
- VRF and PBR are mutually-exclusive on a switch interface. You cannot enable VRF when PBR is enabled on an interface.
- When a policy route map is applied on an interface along with ACL and QoS, ACL and QoS have higher precedence.
- IP Source Guard has a higher precedence if a rule matches with the PBR rule on the same interface.
- On an SVI interface, IP Source Guard and PBR rules are merged based on requirements.
- Multicast traffic is not policy-routed. PBR applies only to unicast traffic.

- You can enable PBR on a routed port or an SVI.
- You can define a maximum of 64 IP policy route maps on the switch.
- You can define a maximum of 64 access control entries (ACEs) for PBR on the switch.
- The number of hardware entries used by PBR depends on the route map itself, the ACLs used, and the order of the ACLs and route-map entries. The maximum number of entries is 256.
- VRF and PBR are mutually exclusive on a switch interface. You cannot enable VRF when PBR is enabled on an interface. The reverse is also true, you cannot enable PBR when VRF is enabled on an interface.
- Web Cache Communication Protocol (WCCP) and PBR are mutually exclusive on a switch interface. You cannot enable WCCP when PBR is enabled on an interface. The reverse is also true, you cannot enable PBR when WCCP is enabled on an interface.
- PBR based on TOS, DSCP and IP Precedence are not supported.
- Set interface, set default next-hop and set default interface are not supported.
- **ip next-hop recursive** and **ip next-hop verify availability** features are not available and the next-hop should be directly connected.
- For a single sequence route-map only one set clause is supported at a time. For a route-map with multiple sequences only set clauses of the same type are allowed. For example, if **set ip next-hop** is used in the first sequence, then the second sequence should also have the same set clause **set ip next-hop**.
- Policy-maps with no set actions are supported. Matching packets are routed normally.
- Policy-maps with no match clauses are supported. Set actions are applied to all packets.

The following table summarizes the PBR support for ACL match field options on the switch.

Table 1: PBR Supported ACL Match field Options

Match Field	Supported (Y/N)
Source IP address	Y
Destination IP address	Y
Next Header (ICMP, IGMP, etc.)	N
TCP/UDP Port	N
Type of Service (TOS)	N
Fragmentation Bit	N

This table lists the PBR feature support on the switch.

Table 2: PBR Feature Support

Feature	Support/Scale
PBR on Ingress Traffic	Y
PBR on Egress Traffic	N

Feature	Support/Scale
PBR on Physical Interface (L2 Port)	N
PBR on Physical Interface (Routed Port)	Y
PBR on SVI Interface	Y
PBR on Port Channel (L2)	N
PBR on Port Channel (L3)	N
PBR with VRF	N
Match on IPv4 ACL	Y Note Refer to table above for PBR Supported ACL Match field Options.
Match on Extended/Standard IPv4 ACL	Y
Match Based on Packet Length	N
Match with DENY ACE	N
Action Set Fragment Bit	N
Action to Set Precedence	N
Action to Set Next-Hop	Y
Recursive Next-Hop Action	N
Action to Set Interface	N
Action to Set Default Interface	N
Action to Set IP Precedence	Y
Action to Set IP VRF	Y
Set IP default Next-Hop	N
Set IP Default VRF	N
PBR on Multicast Traffic	N
PBR on IPv6 Traffic	N
Route-Map Deny	N
MAX number of route-map supported	64
MAX number of ACL policies supported	64
Local PBR	Y

How to Configure PBR

By default, PBR is disabled on the switch. To enable PBR, you must create a route map that specifies the match criteria and the resulting action. Then, you must enable PBR for that route map on an interface. All packets arriving on the specified interface matching the match clauses are subject to PBR.

Before you begin

Refer to [Restrictions and Limitations for Policy-Based Routing](#), on page 16.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	route-map <i>map-tag</i> [permit] [<i>sequence number</i>] Example: <pre>Device(config)# route-map pbr-map permit</pre>	Defines route maps that are used to control where packets are output, and enters route-map configuration mode. <ul style="list-style-type: none"> • <i>map-tag</i> – A meaningful name for the route map. The ip policy route-map interface configuration command uses this name to reference the route map. Multiple route-map statements with the same map tag define a single route map. • (Optional) permit – If permit is specified and the match criteria are met for this route map, the route is policy routed as defined by the set actions. • (Optional) <i>sequence number</i> – The sequence number shows the position of the route-map statement in the given route map.
Step 4	match ip address { <i>access-list-number</i> <i>access-list-name</i> } [<i>access-list-number</i> ... <i>access-list-name</i>] Example: <pre>Device(config-route-map)# match ip address 110 140</pre>	Matches the source and destination IP addresses that are permitted by one or more standard or extended access lists. ACLs can match on more than one source and destination IP address. If you do not specify a match command, the route map is applicable to all packets.
Step 5	set ip next-hop <i>ip-address</i> [... <i>ip-address</i>] Example: <pre>Device(config-route-map)# set ip next-hop 10.1.6.2</pre>	Specifies the action to be taken on the packets that match the criteria. Sets next hop to which to route the packet (the next hop must be adjacent).

	Command or Action	Purpose
Step 6	set ip vrf vrf-name next-hop ip-address [...ip-address] Example: <pre>Device(config-route-map)# set ip vrf myvrf next-hop 10.5.5.5</pre>	Allows you to apply policy-based routing to a VRF interface.
Step 7	set ip precedence [number / name] Example: <pre>Device(config-route-map)# set ip precedence 5</pre>	<ul style="list-style-type: none"> • 0—routine • 1—priority • 2—immediate • 3—flash • 4—flash-override • 5—critical • 6—internet • 7—network <p>Sets the precedence value in the IP header.</p>
Step 8	exit Example: <pre>Device(config-route-map)# exit</pre>	Returns to global configuration mode.
Step 9	interface interface-id Example: <pre>Device(config)# interface gigabitethernet 1/1</pre>	Enters interface configuration mode, and specifies the interface to be configured.
Step 10	ip policy route-map map-tag Example: <pre>Device(config-if)# ip policy route-map pbr-map</pre>	Enables PBR on a Layer 3 interface, and identify the route map to use. You can configure only one route map on an interface. However, you can have multiple route map entries with different sequence numbers. These entries are evaluated in the order of sequence number until the first match. If there is no match, packets are routed as usual.
Step 11	exit Example: <pre>Device(config-if)# exit</pre>	Returns to global configuration mode.
Step 12	ip local policy route-map map-tag Example: <pre>Device(config)# ip local policy route-map local-pbr</pre>	(Optional) Enables local PBR to perform policy-based routing on packets originating at the switch. This applies to packets generated by the switch, and not to incoming packets.
Step 13	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 14	show route-map [<i>map-name</i>] Example: Device# show route-map	(Optional) Displays all the route maps configured or only the one specified to verify configuration.
Step 15	show ip policy Example: Device# show ip policy	(Optional) Displays policy route maps attached to the interface.
Step 16	show ip local policy Example: Device# show ip local policy	(Optional) Displays whether or not local policy routing is enabled and, if so, the route map being used.



CHAPTER 3

Configuring IPv6 Unicast Routing

- [Information About Configuring IPv6 Unicast Routing, on page 23](#)
- [How to Configure IPv6 Unicast Routing, on page 31](#)
- [Displaying IPv6, on page 52](#)
- [Configuration Examples for IPv6 Unicast Routing, on page 52](#)
- [Additional References, on page 56](#)

Information About Configuring IPv6 Unicast Routing

This chapter describes how to configure IPv6 unicast routing on the switch.



Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Understanding IPv6

IPv4 users can move to IPv6 and receive services such as end-to-end security, quality of service (QoS), and globally unique addresses. The IPv6 address space reduces the need for private addresses and Network Address Translation (NAT) processing by border routers at network edges.

For information about how Cisco Systems implements IPv6, go to:

http://www.cisco.com/en/US/products/ps6553/products_ios_technology_home.html

For information about IPv6 and other features in this chapter

- See the *Cisco IOS IPv6 Configuration Library*.
- Use the Search field on Cisco.com to locate the Cisco IOS software documentation. For example, if you want information about static routes, you can enter *Implementing Static Routes for IPv6* in the search field to learn about static routes.

IPv6 Addresses

The switch supports only IPv6 unicast addresses. It does not support site-local unicast addresses, or anycast addresses.

The IPv6 128-bit addresses are represented as a series of eight 16-bit hexadecimal fields separated by colons in the format: n:n:n:n:n:n:n:n. This is an example of an IPv6 address:

```
2031:0000:130F:0000:0000:09C0:080F:130B
```

For easier implementation, leading zeros in each field are optional. This is the same address without leading zeros:

```
2031:0:130F:0:0:9C0:80F:130B
```

You can also use two colons (::) to represent successive hexadecimal fields of zeros, but you can use this short version only once in each address:

```
2031:0:130F::09C0:080F:130B
```

For more information about IPv6 address formats, address types, and the IPv6 packet header, see the http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xr-3e/ip6b-xe-3e-book.html of *Cisco IOS IPv6 Configuration Library* on Cisco.com.

- IPv6 Address Formats
- IPv6 Address Type: Multicast
- IPv6 Address Output Display
- Simplified IPv6 Packet Header

Supported IPv6 Unicast Routing Features

These sections describe the IPv6 protocol features supported by the switch:

128-Bit Wide Unicast Addresses

The switch supports aggregatable global unicast addresses and link-local unicast addresses. It does not support site-local unicast addresses.

- Aggregatable global unicast addresses are IPv6 addresses from the aggregatable global unicast prefix. The address structure enables strict aggregation of routing prefixes and limits the number of routing table entries in the global routing table. These addresses are used on links that are aggregated through organizations and eventually to the Internet service provider.

These addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Current global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). Addresses with a prefix of 2000::/3(001) through E000::/3(111) must have 64-bit interface identifiers in the extended unique identifier (EUI)-64 format.

- Link local unicast addresses can be automatically configured on any interface by using the link-local prefix FE80::/10(1111 1110 10) and the interface identifier in the modified EUI format. Link-local addresses are used in the neighbor discovery protocol (NDP) and the stateless autoconfiguration process. Nodes on a local link use link-local addresses and do not require globally unique addresses to communicate. IPv6 routers do not forward packets with link-local source or destination addresses to other links.

For more information, see the section about IPv6 unicast addresses in the “Implementing IPv6 Addressing and Basic Connectivity” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

DNS for IPv6

IPv6 supports Domain Name System (DNS) record types in the DNS name-to-address and address-to-name lookup processes. The DNS AAAA resource record types support IPv6 addresses and are equivalent to an A address record in IPv4. The switch supports DNS resolution for IPv4 and IPv6.

Path MTU Discovery for IPv6 Unicast

The switch supports advertising the system maximum transmission unit (MTU) to IPv6 nodes and path MTU discovery. Path MTU discovery allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path. In IPv6, if a link along the path is not large enough to accommodate the packet size, the source of the packet handles the fragmentation.

ICMPv6

The Internet Control Message Protocol (ICMP) in IPv6 generates error messages, such as ICMP destination unreachable messages, to report errors during processing and other diagnostic functions. In IPv6, ICMP packets are also used in the neighbor discovery protocol and path MTU discovery.

Neighbor Discovery

The switch supports NDP for IPv6, a protocol running on top of ICMPv6, and static neighbor entries for IPv6 stations that do not support NDP. The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), to verify the reachability of the neighbor, and to keep track of neighboring routers.

The switch supports ICMPv6 redirect for routes with mask lengths less than 64 bits. ICMP redirect is not supported for host routes or for summarized routes with mask lengths greater than 64 bits.

Neighbor discovery throttling ensures that the switch CPU is not unnecessarily burdened while it is in the process of obtaining the next hop forwarding information to route an IPv6 packet. The switch drops any additional IPv6 packets whose next hop is the same neighbor that the switch is actively trying to resolve. This drop avoids further load on the CPU.

Default Router Preference

The switch supports IPv6 default router preference (DRP), an extension in router advertisement messages. DRP improves the ability of a host to select an appropriate router, especially when the host is multihomed and the routers are on different links. The switch does not support the Route Information Option in RFC 4191.

An IPv6 host maintains a default router list from which it selects a router for traffic to offlink destinations. The selected router for a destination is then cached in the destination cache. NDP for IPv6 specifies that routers that are reachable or probably reachable are preferred over routers whose reachability is unknown or suspect. For reachable or probably reachable routers, NDP can either select the same router every time or cycle through the router list. By using DRP, you can configure an IPv6 host to prefer one router over another, provided both are reachable or probably reachable.

For configuring DRP for IPv6, see the *Configuring Default Router Preference* section.

For more information about DRP for IPv6, see the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

IPv6 Stateless Autoconfiguration and Duplicate Address Detection

The switch uses stateless autoconfiguration to manage link, subnet, and site addressing changes, such as management of host and mobile IP addresses. A host autonomously configures its own link-local address, and booting nodes send router solicitations to request router advertisements for configuring interfaces.

For more information about autoconfiguration and duplicate address detection, see the “Implementing IPv6 Addressing and Basic Connectivity” chapter of *Cisco IOS IPv6 Configuration Library* on Cisco.com.

IPv6 Applications

The switch has IPv6 support for these applications:

- Ping, traceroute, Telnet, and TFTP
- Secure Shell (SSH) over an IPv6 transport
- HTTP server access over IPv6 transport
- DNS resolver for AAAA over IPv4 transport
- Cisco Discovery Protocol (CDP) support for IPv6 addresses

For more information about managing these applications, see the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

DHCP for IPv6 Address Assignment

DHCPv6 enables DHCP servers to pass configuration parameters, such as IPv6 network addresses, to IPv6 clients. The address assignment feature manages non-duplicate address assignment in the correct prefix based on the network where the host is connected. Assigned addresses can be from one or multiple prefix pools. Additional options, such as default domain and DNS name-server address, can be passed back to the client. Address pools can be assigned for use on a specific interface, on multiple interfaces, or the server can automatically find the appropriate pool.

For configuring DHCP for IPv6, see the *Configuring DHCP for IPv6 Address Assignment* section.

For more information about configuring the DHCPv6 client, server, or relay agent functions, see the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Static Routes for IPv6

Static routes are manually configured and define an explicit route between two networking devices. Static routes are useful for smaller networks with only one path to an outside network or to provide security for certain types of traffic in a larger network.

Configuring Static Routing for IPv6 (CLI)

For configuring static routes for IPv6, see the *Configuring Static Routing for IPv6* section.

For more information about static routes, see the “Implementing Static Routes for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Policy-Based Routing for IPv6

Policy-based routing (PBR) gives you a flexible means of routing packets by allowing you to configure a defined policy for traffic flows, which lessens reliance on routes derived from routing protocols. Therefore, PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to set the IPv6 precedence. For a simple policy, you can use any one

of these tasks; for a complex policy, you can use all of them. It also allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.

PBR for IPv6 may be applied to both forwarded and originated IPv6 packets. For forwarded packets, PBR for IPv6 will be implemented as an IPv6 input interface feature, supported in the following forwarding paths:

- Process
- Cisco Express Forwarding (formerly known as CEF)
- Distributed Cisco Express Forwarding

Policies can be based on the IPv6 address, port numbers, protocols, or packet size.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists, then, establish the match criteria.
- Set IPv6 precedence bits, giving the network the ability to enable differentiated classes of service.
- Route packets to specific traffic-engineered paths; you might need to route them to allow a specific quality of service (QoS) through the network.

PBR allows you to classify and mark packets at the edge of the network. PBR marks a packet by setting precedence value. The precedence value can be used directly by devices in the network core to apply the appropriate QoS to a packet, which keeps packet classification at your network edge.

For enabling PBR for IPv6, see the *Enabling Local PBR for IPv6* section.

For enabling IPv6 PBR for an interface, see the *Enabling IPv6 PBR on an Interface* section.

RIP for IPv6

Routing Information Protocol (RIP) for IPv6 is a distance-vector protocol that uses hop count as a routing metric. It includes support for IPv6 addresses and prefixes and the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages.

For configuring RIP for IPv6, see the *Configuring RIP for IPv6* section.

For more information about RIP for IPv6, see the “Implementing RIP for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

OSPF for IPv6

The switch supports Open Shortest Path First (OSPF) for IPv6, a link-state protocol for IP.

For configuring OSPF for IPv6, see the *Configuring OSPF for IPv6* section.

For more information, see *Cisco IOS IPv6 Configuration Library* on Cisco.com.

EIGRP IPv6

Switches support the Enhanced Interior Gateway Routing Protocol (EIGRP) for IPv6. It is configured on the interfaces on which it runs and does not require a global IPv6 address. Switches running Network Essentials only support EIGRPv6 stub routing.

Before running, an instance of EIGRP IPv6 requires an implicit or explicit router ID. An implicit router ID is derived from a local IPv6 address, so any IPv6 node always has an available router ID. However, EIGRP IPv6 might be running in a network with only IPv6 nodes and therefore might not have an available IPv6 router ID.

For configuring EIGRP for IPv6, see the *Configuring EIGRP for IPv6* section.

For more information about EIGRP for IPv6, see the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

EIGRPv6 Stub Routing

The EIGRPv6 stub routing feature, reduces resource utilization by moving routed traffic closer to the end user.

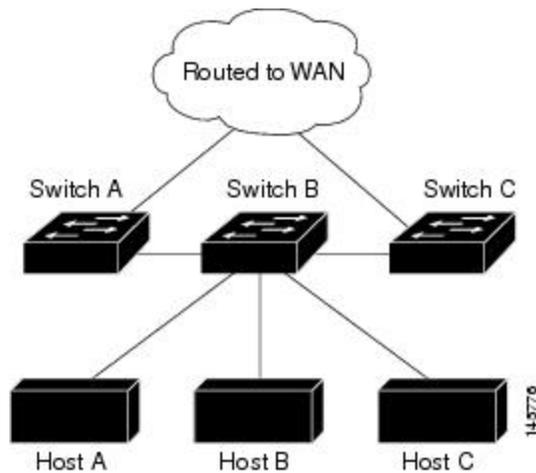
In a network using EIGRPv6 stub routing, the only allowable route for IPv6 traffic to the user is through a switch that is configured with EIGRPv6 stub routing. The switch sends the routed traffic to interfaces that are configured as user interfaces or are connected to other devices.

When using EIGRPv6 stub routing, you need to configure the distribution and remote routers to use EIGRPv6 and to configure only the switch as a stub. Only specified routes are propagated from the switch. The switch responds to all queries for summaries, connected routes, and routing updates.

Any neighbor that receives a packet informing it of the stub status does not query the stub router for any routes, and a router that has a stub peer does not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

In the figure given below, switch B is configured as an EIGRPv6 stub router. Switches A and C are connected to the rest of the WAN. Switch B advertises connected, static, redistribution, and summary routes to switch A and C. Switch B does not advertise any routes learned from switch A (and the reverse).

Figure 3: EIGRP Stub Router Configuration



For more information about EIGRPv6 stub routing, see “Implementing EIGRP for IPv6” section of the *Cisco IOS IP Configuration Guide, Volume 2 of 3: Routing Protocols, Release 12.4*.

SNMP and Syslog Over IPv6

To support both IPv4 and IPv6, IPv6 network management requires both IPv6 and IPv4 transports. Syslog over IPv6 supports address data types for these transports.

Simple Network Management Protocol (SNMP) and syslog over IPv6 provide these features:

- Support for both IPv4 and IPv6
- IPv6 transport for SNMP and to modify the SNMP agent to support traps for an IPv6 host
- SNMP- and syslog-related MIBs to support IPv6 addressing

- Configuration of IPv6 hosts as trap receivers

For support over IPv6, SNMP modifies the existing IP transport mapping to simultaneously support IPv4 and IPv6. These SNMP actions support IPv6 transport management:

- Opens User Datagram Protocol (UDP) SNMP socket with default settings
- Provides a new transport mechanism called *SR_IPV6_TRANSPORT*
- Sends SNMP notifications over IPv6 transport
- Supports SNMP-named access lists for IPv6 transport
- Supports SNMP proxy forwarding using IPv6 transport
- Verifies SNMP Manager feature works with IPv6 transport

For information on SNMP over IPv6, including configuration procedures, see the “Managing Cisco IOS Applications over IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

For information about syslog over IPv6, including configuration procedures, see the “Implementing IPv6 Addressing and Basic Connectivity” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

HTTP(S) Over IPv6

The HTTP client sends requests to both IPv4 and IPv6 HTTP servers, which respond to requests from both IPv4 and IPv6 HTTP clients. URLs with literal IPv6 addresses must be specified in hexadecimal using 16-bit values between colons.

The accept socket call chooses an IPv4 or IPv6 address family. The accept socket is either an IPv4 or IPv6 socket. The listening socket continues to listen for both IPv4 and IPv6 signals that indicate a connection. The IPv6 listening socket is bound to an IPv6 wildcard address.

The underlying TCP/IP stack supports a dual-stack environment. HTTP relies on the TCP/IP stack and the sockets for processing network-layer interactions.

Basic network connectivity (**ping**) must exist between the client and the server hosts before HTTP connections can be made.

For more information, see the “Managing Cisco IOS Applications over IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Unsupported IPv6 Unicast Routing Features

The switch does not support these IPv6 features:

- IPv6 packets that are destined to site-local addresses
- Tunneling protocols, such as IPv4-to-IPv6 or IPv6-to-IPv4
- The switch as a tunnel endpoint supporting IPv4-to-IPv6 or IPv6-to-IPv4 tunneling protocols
- IPv6 Web Cache Communication Protocol (WCCP)

IPv6 Feature Limitations

Because IPv6 is implemented in switch hardware, some limitations occur due to the IPv6 compressed addresses in the hardware memory. These hardware limitations result in some loss of functionality and limits some features.

These are feature limitations.

- The switch cannot apply QoS classification on source-routed IPv6 packets in hardware.

IPv6 and Switch Stacks

The switch supports IPv6 forwarding across the stack and IPv6 host functionality on the active switch. The active switch runs the IPv6 unicast routing protocols and computes the routing tables. They receive the tables and create hardware IPv6 routes for forwarding. The active switch also runs all IPv6 applications.

If a new switch becomes the active switch, it recomputes the IPv6 routing tables and distributes them to the member switches. While the new active switch is being elected and is resetting, the switch stack does not forward IPv6 packets. The stack MAC address changes, which also change the IPv6 address. When you specify the stack IPv6 address with an extended unique identifier (EUI) by using the **ipv6 address ipv6-prefix/prefix length eui-64** interface configuration command, the address is based on the interface MAC address. See the *Configuring IPv6 Addressing and Enabling IPv6 Routing* section.

If you configure the persistent MAC address feature on the stack and the active switch changes, the stack MAC address does not change for approximately 4 minutes.

These are the functions of IPv6 active switch and members:

- Active switch:
 - runs IPv6 routing protocols
 - generates routing tables
 - distributes routing tables to member switches that use distributed Cisco Express Forwarding for IPv6
 - runs IPv6 host functionality and IPv6 applications
- Member switch:
 - receives Cisco Express Forwarding for IPv6 routing tables from the active switch
 - programs the routes into hardware



Note IPv6 packets are routed in hardware across the stack if the packet does not have exceptions (IPv6 Options) and the switches in the stack have not run out of hardware resources.

- flushes the Cisco Express Forwarding for IPv6 tables on active switch re-election

Default IPv6 Configuration

Table 3: Default IPv6 Configuration

Feature	Default Setting
IPv6 routing	Disabled globally and on all interfaces Note IP routing is enabled by default , but IPv6 unicast-routing must be enabled by the user.
Cisco Express Forwarding for IPv6 or distributed Cisco Express Forwarding for IPv6	Disabled (IPv4 Cisco Express Forwarding and distributed Cisco Express Forwarding are enabled by default) Note When IPv6 routing is enabled, Cisco Express Forwarding for IPv6 and distributed Cisco Express Forwarding for IPv6 are automatically enabled.
IPv6 addresses	None configured

How to Configure IPv6 Unicast Routing

The following sections shows the various configuration options available for IPv6 Unicast Routing

Configuring IPv6 Addressing and Enabling IPv6 Routing

This section describes how to assign IPv6 addresses to individual Layer 3 interfaces and to globally forward IPv6 traffic on the switch.

Before configuring IPv6 on the switch, consider these guidelines:

- Not all features discussed in this chapter are supported by the switch. See the [Unsupported IPv6 Unicast Routing Features](#).
- In the **ipv6 address** interface configuration command, you must enter the *ipv6-address* and *ipv6-prefix* variables with the address specified in hexadecimal using 16-bit values between colons. The *prefix-length* variable (preceded by a slash [/]) is a decimal value that shows how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address).

To forward IPv6 traffic on an interface, you must configure a global IPv6 address on that interface. Configuring an IPv6 address on an interface automatically configures a link-local address and activates IPv6 for the interface. The configured interface automatically joins these required multicast groups for that link:

- solicited-node multicast group FF02:0:0:0:1:ff00::/104 for each unicast address assigned to the interface (this address is used in the neighbor discovery process.)
- all-nodes link-local multicast group FF02::1
- all-routers link-local multicast group FF02::2

To remove an IPv6 address from an interface, use the **no ipv6 address *ipv6-prefix/prefix length eui-64*** or **no ipv6 address *ipv6-address link-local*** interface configuration command. To remove all manually configured IPv6 addresses from an interface, use the **no ipv6 address** interface configuration command without arguments. To disable IPv6 processing on an interface that has not been explicitly configured with an IPv6 address, use the **no ipv6 enable** interface configuration command. To globally disable IPv6 routing, use the **no ipv6 unicast-routing** global configuration command.

For more information about configuring IPv6 routing, see the “Implementing Addressing and Basic Connectivity for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

To assign an IPv6 address to a Layer 3 interface and enable IPv6 routing, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <code>> enable</code>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <code># configure terminal</code>	Enters global configuration mode.
Step 3	end Example: <code>(config)# end</code>	Returns to privileged EXEC mode.
Step 4	reload Example: <code># reload</code>	Reloads the operating system.
Step 5	configure terminal Example: <code># configure terminal</code>	Enters global configuration mode after the switch reloads.
Step 6	interface <i>interface-id</i> Example: <code>(config)# interface gigabitethernet 1/0/1</code>	Enters interface configuration mode, and specifies the Layer 3 interface to configure. The interface can be a physical interface, a switch virtual interface (SVI), or a Layer 3 EtherChannel.

	Command or Action	Purpose
Step 7	no switchport Example: <pre>(config-if)# no switchport</pre>	Removes the interface from Layer 2 configuration mode (if it is a physical interface).
Step 8	Use one of the following: <ul style="list-style-type: none"> • ipv6 address <i>ipv6-prefix/prefix length eui-64</i> • ipv6 address <i>ipv6-address/prefix length</i> • ipv6 address <i>ipv6-address link-local</i> • ipv6 enable • ipv6 address <i>WORD</i> • ipv6 address <i>autoconfig</i> • ipv6 address <i>dhcp</i> Example: <pre>(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64</pre> <pre>(config-if)# ipv6 address 2001:0DB8:c18:1::/64</pre> <pre>(config-if)# ipv6 address 2001:0DB8:c18:1:: link-local</pre> <pre>(config-if)# ipv6 enable</pre>	<ul style="list-style-type: none"> • Specifies a global IPv6 address with an extended unique identifier (EUI) in the low-order 64 bits of the IPv6 address. Specify only the network prefix; the last 64 bits are automatically computed from the switch MAC address. This enables IPv6 processing on the interface. • Manually configures an IPv6 address on the interface. • Specifies a link-local address on the interface to be used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface. This command enables IPv6 processing on the interface. • Automatically configures an IPv6 link-local address on the interface, and enables the interface for IPv6 processing. The link-local address can only be used to communicate with nodes on the same link.
Step 9	exit Example: <pre>(config-if)# exit</pre>	Returns to global configuration mode.
Step 10	ipv6 unicast-routing Example: <pre>(config)# ipv6 unicast-routing</pre>	Enables forwarding of IPv6 unicast data packets.
Step 11	end Example: <pre>(config)# end</pre>	Returns to privileged EXEC mode.
Step 12	show ipv6 interface interface-id Example:	Verifies your entries.

	Command or Action	Purpose
	<code># show ipv6 interface gigabitethernet 1/0/1</code>	
Step 13	copy running-config startup-config Example: <code># copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring IPv4 and IPv6 Protocol Stacks

To configure a Layer 3 interface to support both IPv4 and IPv6 and to enable IPv6 routing, perform this procedure:



Note To disable IPv6 processing on an interface that has not been configured with an IPv6 address, use the **no ipv6 enable** interface configuration command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **ipv6 unicast-routing**
5. **interface** *interface-id*
6. **no switchport**
7. **ip address** *ip-address mask* [**secondary**]
8. Use one of the following:
 - **ipv6 address** *ipv6-prefix/prefix length eui-64*
 - **ipv6 address** *ipv6-address/prefix length*
 - **ipv6 address** *ipv6-address link-local*
 - **ipv6 enable**
9. **end**
10. Use one of the following:
 - **show interface** *interface-id*
 - **show ip interface** *interface-id*
 - **show ipv6 interface** *interface-id*
11. **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	Purpose
	Example: <pre>> enable</pre>	Enter your password if prompted.
Step 2	configure terminal Example: <pre># configure terminal</pre>	Enters global configuration mode.
Step 3	ip routing Example: <pre>(config)# ip routing</pre>	Enables routing on the switch.
Step 4	ipv6 unicast-routing Example: <pre>(config)# ipv6 unicast-routing</pre>	Enables forwarding of IPv6 data packets on the switch.
Step 5	interface <i>interface-id</i> Example: <pre>(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 6	no switchport Example: <pre>(config-if)# no switchport</pre>	Removes the interface from Layer 2 configuration mode (if it is a physical interface).
Step 7	ip address <i>ip-address mask</i> [secondary] Example: <pre>(config-if)# ip address 10.1.2.3 255.255.255</pre>	Specifies a primary or secondary IPv4 address for the interface.
Step 8	Use one of the following: <ul style="list-style-type: none"> • ipv6 address <i>ipv6-prefix/prefix length eui-64</i> • ipv6 address <i>ipv6-address/prefix length</i> • ipv6 address <i>ipv6-address link-local</i> • ipv6 enable 	<ul style="list-style-type: none"> • Specifies a global IPv6 address. Specify only the network prefix; the last 64 bits are automatically computed from the switch MAC address. • Specifies a link-local address on the interface to be used instead of the automatically configured link-local address when IPv6 is enabled on the interface. • Automatically configures an IPv6 link-local address on the interface, and enables the interface for IPv6 processing. The link-local address can only be used to communicate with nodes on the same link.

	Command or Action	Purpose
		Note To remove all manually configured IPv6 addresses from an interface, use the no ipv6 address interface configuration command without arguments.
Step 9	end Example: <code>(config)# end</code>	Returns to privileged EXEC mode.
Step 10	Use one of the following: <ul style="list-style-type: none"> • show interface <i>interface-id</i> • show ip interface <i>interface-id</i> • show ipv6 interface <i>interface-id</i> 	Verifies your entries.
Step 11	copy running-config startup-config Example: <code># copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring Default Router Preference

Router advertisement messages are sent with the default router preference (DRP) configured by the **ipv6 nd router-preference** interface configuration command. If no DRP is configured, RAs are sent with a medium preference.

A DRP is useful when two routers on a link might provide equivalent, but not equal-cost routing, and policy might dictate that hosts should prefer one of the routers.

For more information about configuring DRP for IPv6, see the “Implementing IPv6 Addresses and Basic Connectivity” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

To configure a DRP for a router on an interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <code>> enable</code>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <code># configure terminal</code>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>interface-id</i> Example: <pre>(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode and identifies the Layer 3 interface on which you want to specify the DRP.
Step 4	ipv6 nd router-preference {high medium low} Example: <pre>(config-if)# ipv6 nd router-preference medium</pre>	Specifies a DRP for the router on the switch interface.
Step 5	end Example: <pre>(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	show ipv6 interface Example: <pre># show ipv6 interface</pre>	Verifies the configuration.
Step 7	copy running-config startup-config Example: <pre># copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring IPv6 ICMP Rate Limiting

ICMP rate limiting is enabled by default with a default interval between error messages of 100 milliseconds and a bucket size (maximum number of tokens to be stored in a bucket) of 10.

To change the ICMP rate-limiting parameters, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	<code># configure terminal</code>	
Step 3	<p>ipv6 icmp error-interval <i>interval</i> [<i>bucketsize</i>]</p> <p>Example:</p> <pre>(config)# ipv6 icmp error-interval 50 20</pre>	<p>Configures the interval and bucket size for IPv6 ICMP error messages:</p> <ul style="list-style-type: none"> • <i>interval</i>—The interval (in milliseconds) between tokens being added to the bucket. The range is from 0 to 2147483647 milliseconds. • <i>bucketsize</i>—(Optional) The maximum number of tokens stored in the bucket. The range is from 1 to 200.
Step 4	<p>end</p> <p>Example:</p> <pre>(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	<p>show ipv6 interface [<i>interface-id</i>]</p> <p>Example:</p> <pre># show ipv6 interface gigabitethernet0/1</pre>	Verifies your entries.
Step 6	<p>copy running-config startup-config</p> <p>Example:</p> <pre># copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Cisco Express Forwarding and distributed Cisco Express Forwarding for IPv6

Cisco Express Forwarding is a Layer 3 IP switching technology to improve network performance. Cisco Express Forwarding implements an advanced IP look-up and forwarding algorithm to deliver maximum Layer 3 switching performance. It is less CPU-intensive than fast-switching route-caching, allowing more CPU processing power to be dedicated to packet forwarding. In a switch stack, the hardware uses distributed Cisco Express Forwarding in the stack. IPv4 Cisco Express Forwarding and distributed Cisco Express Forwarding are enabled by default. IPv6 Cisco Express Forwarding and distributed Cisco Express Forwarding are disabled by default, but automatically enabled when you configure IPv6 routing.

IPv6 Cisco Express Forwarding and distributed Cisco Express Forwarding are automatically disabled when IPv6 routing is unconfigured. IPv6 Cisco Express Forwarding and distributed Cisco Express Forwarding cannot be disabled through configuration. You can verify the IPv6 state by entering the **show ipv6 cef** privileged EXEC command.

To route IPv6 unicast packets, you must first globally configure forwarding of IPv6 unicast packets by using the **ipv6 unicast-routing** global configuration command, and you must configure an IPv6 address and IPv6 processing on an interface by using the **ipv6 address** interface configuration command.

For more information about configuring Cisco Express Forwarding and distributed Cisco Express Forwarding, see *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Configuring Static Routing for IPv6

For more information about configuring static IPv6 routing, see the “Implementing Static Routes for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

To configure static IPv6 routing, perform this procedure:

Before you begin

You must enable routing by using the **ip routing** global configuration command, enable the forwarding of IPv6 packets by using the **ipv6 unicast-routing** global configuration command, and enable IPv6 on at least one Layer 3 interface by configuring an IPv6 address on the interface.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre># configure terminal</pre>	Enters global configuration mode.
Step 3	ipv6 route <i>ipv6-prefix/prefix length {ipv6-address interface-id [ipv6-address]} [administrative distance]</i> Example: <pre>(config)# ipv6 route 2001:0DB8::/32 gigabitethernet2/0/1 130</pre>	Configures a static IPv6 route. <ul style="list-style-type: none"> • <i>ipv6-prefix</i>—The IPv6 network that is the destination of the static route. It can also be a hostname when static host routes are configured. • <i>/prefix length</i>—The length of the IPv6 prefix. A decimal value that shows 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. • <i>ipv6-address</i>—The IPv6 address of the next hop that can be used to reach the specified network. The IPv6 address of the next hop need not be directly connected; recursion is done to find the IPv6 address of the directly connected next hop. The address must be in

	Command or Action	Purpose
		<p>the form documented in RFC 2373, specified in hexadecimal using 16-bit values between colons.</p> <ul style="list-style-type: none"> • <i>interface-id</i>—Specifies direct static routes from point-to-point and broadcast interfaces. With point-to-point interfaces, there is no need to specify the IPv6 address of the next hop. With broadcast interfaces, you should always specify the IPv6 address of the next hop, or ensure that the specified prefix is assigned to the link, specifying a link-local address as the next hop. You can optionally specify the IPv6 address of the next hop to which packets are sent. <p>Note You must specify an <i>interface-id</i> when using a link-local address as the next hop (the link-local next hop must also be an adjacent router).</p> <ul style="list-style-type: none"> • <i>administrative distance</i>—(Optional) An administrative distance. The range is 1 to 254; the default value is 1, which gives static routes precedence over any other type of route except connected routes. To configure a floating static route, use an administrative distance greater than that of the dynamic routing protocol.
Step 4	<p>end</p> <p>Example:</p> <pre>(config)# end</pre>	Returns to privileged EXEC mode.
Step 5	<p>Use one of the following:</p> <ul style="list-style-type: none"> • show ipv6 static [<i>ipv6-address</i> <i>ipv6-prefix/prefix length</i>] [interface <i>interface-id</i>] [detail][recursive] [detail] • show ipv6 route static [<i>updated</i>] <p>Example:</p> <pre># show ipv6 static 2001:0DB8::/32 interface gigabitethernet2/0/1</pre> <p>or</p> <pre># show ipv6 route static</pre>	<p>Verifies your entries by displaying the contents of the IPv6 routing table.</p> <ul style="list-style-type: none"> • interface <i>interface-id</i>—(Optional) Displays only those static routes with the specified interface as an egress interface. • recursive—(Optional) Displays only recursive static routes. The recursive keyword is mutually exclusive with the interface keyword, but it can be used with or without the IPv6 prefix included in the command syntax. • detail—(Optional) Displays this additional information: <ul style="list-style-type: none"> • For valid recursive routes, the output path set, and maximum resolution depth. • For invalid routes, the reason why the route is not valid.

	Command or Action	Purpose
Step 6	copy running-config startup-config Example: <pre># copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Enabling IPv6 PBR on an Interface

To enable Policy-Based Routing (PBR) for IPv6, you must create a route map that specifies the packet match criteria and desired policy-route action. Then you associate the route map on the required interface. All packets arriving on the specified interface that match the match clauses will be subject to PBR.

In PBR, the **set vrf** command decouples the virtual routing and forwarding (VRF) instance and interface association and allows the selection of a VRF based on access control list (ACL)-based classification using existing PBR or route-map configurations. It provides a single router with multiple routing tables and the ability to select routes based on ACL classification. The router classifies packets based on ACL, selects a routing table, looks up the destination address, and then routes the packet.

To enable PBR for IPv6, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	route-map map-tag [permit deny] [sequence-number] Example: <pre>Device(config)# route-map rip-to-ospf permit</pre>	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing, and enters route-map configuration mode.
Step 4	Do one of the following: <ul style="list-style-type: none"> • match length <i>minimum-length maximum-length</i> • match ipv6 address {<i>prefix-list prefix-list-name access-list-name</i>} Example: <pre>Device(config-route-map)# match length 3 200</pre> Example: <pre>Device(config-route-map)# match ipv6 address marketing</pre>	Specifies the match criteria. <ul style="list-style-type: none"> • You can specify any or all of the following: <ul style="list-style-type: none"> • Matches the Level 3 length of the packet. • Matches a specified IPv6 access list. • If you do not specify a match command, the route map applies to all packets.

	Command or Action	Purpose
Step 5	<p>Do one of the following:</p> <ul style="list-style-type: none"> • set ipv6 next-hop <i>global-ipv6-address</i> [<i>global-ipv6-address...</i>] • set ipv6 default next-hop <i>global-ipv6-address</i> [<i>global-ipv6-address...</i>] <p>Example:</p> <pre>Device(config-route-map)# set ipv6 next-hop 2001:DB8:2003:1::95</pre> <p>Example:</p> <pre>Device(config-route-map)# set ipv6 default next-hop 2001:DB8:2003:1::95</pre>	<p>Specifies the action or actions to take on the packets that match the criteria.</p> <ul style="list-style-type: none"> • You can specify any or all of the following: <ul style="list-style-type: none"> • Sets next hop to which to route the packet (the next hop must be adjacent). • Sets next hop to which to route the packet, if there is no explicit route for this destination.
Step 6	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route-map configuration mode and returns to global configuration mode.
Step 7	<p>interface <i>type number</i></p> <p>Example:</p> <pre>Device(config)# interface FastEthernet 1/0</pre>	Specifies an interface type and number, and places the router in interface configuration mode.
Step 8	<p>ipv6 policy route-map <i>route-map-name</i></p> <p>Example:</p> <pre>Device(config-if)# ipv6 policy-route-map interactive</pre>	Identifies a route map to use for IPv6 PBR on an interface.
Step 9	<p>end</p> <p>Example:</p> <pre>Device(config-if)# end</pre>	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring RIP for IPv6

For more information about configuring RIP routing for IPv6, see the “Implementing RIP for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com,

To configure RIP routing for IPv6, perform this procedure:

Before you begin

Before configuring the switch to run IPv6 RIP, you must enable routing by using the **ip routing** global configuration command, enable the forwarding of IPv6 packets by using the **ipv6 unicast-routing** global configuration command, and enable IPv6 on any Layer 3 interfaces on which IPv6 RIP is to be enabled.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre># configure terminal</pre>	Enters global configuration mode.
Step 3	ipv6 router rip name Example: <pre>(config)# ipv6 router rip cisco</pre>	Configures an IPv6 RIP routing process, and enters router configuration mode for the process.
Step 4	maximum-paths number-paths Example: <pre>(config-router)# maximum-paths 6</pre>	(Optional) Define the maximum number of equal-cost routes that IPv6 RIP can support. The range is from 1 to 32, and the default is 16 routes.
Step 5	exit Example: <pre>(config-router)# exit</pre>	Returns to global configuration mode.
Step 6	interface interface-id Example: <pre>(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 7	ipv6 rip name enable Example: <pre>(config-if)# ipv6 rip cisco enable</pre>	Enables the specified IPv6 RIP routing process on the interface.
Step 8	ipv6 rip name default-information {only originate} Example: <pre>(config-if)# ipv6 rip cisco default-information</pre>	(Optional) Originates the IPv6 default route (::/0) into the RIP routing process updates sent from the specified interface.

	Command or Action	Purpose
	<code>only</code>	<p>Note To avoid routing loops after the IPv6 default route (::/0) is originated from any interface, the routing process ignores all default routes received on any interface.</p> <ul style="list-style-type: none"> • only—Select to originate the default route, but suppress all other routes in the updates sent on this interface. • originate—Select to originate the default route in addition to all other routes in the updates sent on this interface.
Step 9	<p><code>end</code></p> <p>Example:</p> <pre>(config)# end</pre>	Returns to privileged EXEC mode.
Step 10	<p>Use one of the following:</p> <ul style="list-style-type: none"> • <code>show ipv6 rip [name] [interface interface-id] [database] [next-hops]</code> • <code>show ipv6 rip</code> <p>Example:</p> <pre># show ipv6 rip cisco interface gigabitethernet 2/0/1</pre> <p>or</p> <pre># show ipv6 rip</pre>	<ul style="list-style-type: none"> • Displays information about current IPv6 RIP processes. • Displays the current contents of the IPv6 routing table.
Step 11	<p><code>copy running-config startup-config</code></p> <p>Example:</p> <pre># copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring OSPF for IPv6

For more information about configuring OSPF routing for IPv6, see the “Implementing OSPF for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

To configure OSPF routing for IPv6, perform this procedure:

Before you begin

You can customize OSPF for IPv6 for your network. However, the defaults for OSPF in IPv6 are set to meet the requirements of most customers and features.

Follow these guidelines:

- Be careful when changing the defaults for IPv6 commands. Changing the defaults might adversely affect OSPF for the IPv6 network.
- Before you enable IPv6 OSPF on an interface, you must enable routing by using the **ip routing** global configuration command, enable the forwarding of IPv6 packets by using the **ipv6 unicast-routing** global configuration command, and enable IPv6 on Layer 3 interfaces on which you are enabling IPv6 OSPF.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre># configure terminal</pre>	Enters global configuration mode.
Step 3	ipv6 router ospf process-id Example: <pre>(config)# ipv6 router ospf 21</pre>	Enables OSPF router configuration mode for the process. The process ID is the number assigned administratively when enabling the OSPF for IPv6 routing process. It is locally assigned and can be a positive integer from 1 to 65535.
Step 4	area area-id range {ipv6-prefix/prefix length} [advertise not-advertise] [cost cost] Example: <pre>(config)# area .3 range 2001:0DB8::/32 not-advertise</pre>	(Optional) Consolidates and summarizes routes at an area boundary. <ul style="list-style-type: none"> • area-id—Identifier of the area about which routes are to be summarized. It can be specified as either a decimal value or as an IPv6 prefix. • ipv6-prefix/prefix length—The destination IPv6 network and a decimal value that shows 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. • advertise—(Optional) Sets the address range status to advertise and generate a Type 3 summary link-state advertisement (LSA). • not-advertise—(Optional) Sets the address range status to DoNotAdvertise. The Type 3 summary LSA is suppressed, and component networks remain hidden from other networks.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • cost <i>cost</i>—(Optional) Sets the metric or cost for this summary route, which is used during OSPF SPF calculation to determine the shortest paths to the destination. The value can be 0 to 16777215.
Step 5	maximum paths <i>number-paths</i> Example: <pre>(config)# maximum paths 16</pre>	(Optional) Defines the maximum number of equal-cost routes to the same destination that IPv6 OSPF should enter in the routing table. The range is from 1 to 32, and the default is 16 paths.
Step 6	exit Example: <pre>(config-if)# exit</pre>	Returns to global configuration mode.
Step 7	interface <i>interface-id</i> Example: <pre>(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 8	router <i>router-id</i> Example: <pre>(config)# router ospfv3 1</pre>	Enters interface configuration mode, and specifies the router to configure.
Step 9	ipv6 ospf <i>process-id</i> area <i>area-id</i> [instance <i>instance-id</i>] Example: <pre>(config-if)# ipv6 ospf 21 area .3</pre>	Enables OSPF for IPv6 on the interface. <ul style="list-style-type: none"> • instance <i>instance-id</i>—(Optional) Instance identifier.
Step 10	end Example: <pre>(config)# end</pre>	Returns to privileged EXEC mode.
Step 11	Use one of the following: <ul style="list-style-type: none"> • show ipv6 ospf [<i>process-id</i>] [<i>area-id</i>] interface [<i>interface-id</i>] • show ipv6 ospf [<i>process-id</i>] [<i>area-id</i>] Example:	<ul style="list-style-type: none"> • Displays information about OSPF interfaces. • Displays general information about OSPF routing processes.

	Command or Action	Purpose
	<pre># show ipv6 ospf 21 interface gigabitethernet2/0/1</pre> <p>or</p> <pre># show ipv6 ospf 21</pre>	
Step 12	<p>copy running-config startup-config</p> <p>Example:</p> <pre># copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring EIGRP for IPv6

Before configuring the switch to run IPv6 EIGRP, enable routing by entering the **ip routing global configuration** command, enable the forwarding of IPv6 packets by entering the **ipv6 unicast-routing global configuration** command, and enable IPv6 on any Layer 3 interfaces on which you want to enable IPv6 EIGRP.

To set an explicit router ID, use the **show ipv6 eigrp** command to see the configured router IDs, and then use the **router-id** command.

As with EIGRP IPv4, you can use EIGRPv6 to specify your EIGRP IPv6 interfaces and to select a subset of those as passive interfaces. Use the **passive-interface** command to make an interface passive, and then use the **no passive-interface** command on selected interfaces to make them active. EIGRP IPv6 does not need to be configured on a passive interface.

For more configuration procedures, see the “Implementing EIGRP for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Configuring IPv6 Unicast Reverse Path Forwarding

The unicast Reverse Path Forwarding (unicast RPF) feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers (ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.



Note

- Do not configure Unicast RPF if the switch is in a mixed hardware stack combining more than one switch type.

For detailed IP unicast RPF configuration information, see the *Other Security Features* chapter in the *Cisco IOS Security Configuration Guide, Release 12.4*.

Configuring DHCP for IPv6 Address Assignment

This section describes only the DHCPv6 address assignment. For more information about configuring the DHCPv6 client, server, or relay agent functions, see the “Implementing DHCP for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Default DHCPv6 Address Assignment Configuration

By default, no DHCPv6 features are configured on the switch.

DHCPv6 Address Assignment Configuration Guidelines

When configuring DHCPv6 address assignment, consider these guidelines:

- In the procedures, the specified interface must be one of these Layer 3 interfaces:
 - DHCPv6 IPv6 routing must be enabled on a Layer 3 interface.
 - SVI: a VLAN interface created by using the **interface vlan *vlan_id*** command.
 - EtherChannel port channel in Layer 3 mode: a port-channel logical interface created by using the **interface port-channel *port-channel-number*** command.
- The switch can act as a DHCPv6 client, server, or relay agent. The DHCPv6 client, server, and relay function are mutually exclusive on an interface.
- The DHCPv6 client, server, or relay agent runs only on the active switch. When there is an active switch re-election, the new active switch retains the DHCPv6 configuration. However, the local RAM copy of the DHCP server database lease information is not retained.

Enabling DHCPv6 Server Function (CLI)

Use the **no** form of the DHCP pool configuration mode commands to change the DHCPv6 pool characteristics. To disable the DHCPv6 server function on an interface, use the **no ipv6 dhcp server** interface configuration command.

To enable the DHCPv6 server function on an interface, perform this procedure:

Procedure

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

	Command or Action	Purpose
Step 3	<p>ipv6 dhcp pool <i>poolname</i></p> <p>Example:</p> <pre>(config)# ipv6 dhcp pool 7</pre>	Enters DHCP pool configuration mode, and define the name for the IPv6 DHCP pool. The pool name can be a symbolic string (such as Engineering) or an integer (such as 0).
Step 4	<p>address prefix <i>IPv6-prefix</i> {lifetime} {<i>t1 t1</i> infinite}</p> <p>Example:</p> <pre>(config-dhcpv6)# address prefix 2001:1000::0/64 lifetime 3600</pre>	<p>(Optional) Specifies an address prefix for address assignment.</p> <p>This address must be in hexadecimal, using 16-bit values between colons.</p> <p>lifetime <i>t1 t1</i>—Specifies a time interval (in seconds) that an IPv6 address prefix remains in the valid state. The range is 5 to 4294967295 seconds. Specify infinite for no time interval.</p>
Step 5	<p>link-address <i>IPv6-prefix</i></p> <p>Example:</p> <pre>(config-dhcpv6)# link-address 2001:1002::0/64</pre>	<p>(Optional) Specifies a link-address IPv6 prefix.</p> <p>When an address on the incoming interface or a link-address in the packet matches the specified IPv6 prefix, the server uses the configuration information pool.</p> <p>This address must be in hexadecimal, using 16-bit values between colons.</p>
Step 6	<p>vendor-specific <i>vendor-id</i></p> <p>Example:</p> <pre>(config-dhcpv6)# vendor-specific 9</pre>	(Optional) Enters vendor-specific configuration mode and specifies a vendor-specific identification number. This number is the vendor IANA Private Enterprise Number. The range is 1 to 4294967295.
Step 7	<p>suboption <i>number</i> {address <i>IPv6-address</i> ascii <i>ASCII-string</i> hex <i>hex-string</i>}</p> <p>Example:</p> <pre>(config-dhcpv6-vs)# suboption 1 address 1000:235D::</pre>	(Optional) Enters a vendor-specific suboption number. The range is 1 to 65535. Enter an IPv6 address, ASCII text, or a hex string as defined by the suboption parameters.
Step 8	<p>exit</p> <p>Example:</p> <pre>(config-dhcpv6-vs)# exit</pre>	Returns to DHCP pool configuration mode.
Step 9	<p>exit</p> <p>Example:</p> <pre>(config-dhcpv6)# exit</pre>	Returns to global configuration mode.

	Command or Action	Purpose
Step 10	interface <i>interface-id</i> Example: <pre>(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the interface to configure.
Step 11	ipv6 dhcp server [<i>poolname</i> automatic] [rapid-commit] [preference value] [allow-hint] Example: <pre>(config-if)# ipv6 dhcp server automatic</pre>	Enables DHCPv6 server function on an interface. <ul style="list-style-type: none"> • poolname—(Optional) User-defined name for the IPv6 DHCP pool. The pool name can be a symbolic string (such as Engineering) or an integer (such as 0). • automatic—(Optional) Enables the system to automatically determine which pool to use when allocating addresses for a client. • rapid-commit—(Optional) Allows two-message exchange method. • preference value—(Optional) Configures the preference value carried in the preference option in the advertise message sent by the server. The range is from 0 to 255. The preference value default is 0. • allow-hint—(Optional) Specifies whether the server should consider client suggestions in the SOLICIT message. By default, the server ignores client hints.
Step 12	end Example: <pre>(config)# end</pre>	Returns to privileged EXEC mode.
Step 13	Do one of the following: <ul style="list-style-type: none"> • show ipv6 dhcp pool • show ipv6 dhcp interface Example: <pre># show ipv6 dhcp pool</pre> or <pre># show ipv6 dhcp interface</pre>	<ul style="list-style-type: none"> • Verifies DHCPv6 pool configuration. • Verifies that the DHCPv6 server function is enabled on an interface.
Step 14	copy running-config startup-config Example: <pre># copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Enabling DHCPv6 Client Function

To enable the DHCPv6 client on an interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>> enable</pre>	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre># configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: <pre>(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the interface to configure.
Step 4	ipv6 address dhcp [rapid-commit] Example: <pre>(config-if)# ipv6 address dhcp rapid-commit</pre>	Enables the interface to acquire an IPv6 address from the DHCPv6 server. rapid-commit —(Optional) Allow two-message exchange method for address assignment.
Step 5	ipv6 dhcp client request [vendor-specific] Example: <pre>(config-if)# ipv6 dhcp client request vendor-specific</pre>	(Optional) Enables the interface to request the vendor-specific option.
Step 6	end Example: <pre>(config)# end</pre>	Returns to privileged EXEC mode.
Step 7	show ipv6 dhcp interface Example: <pre># show ipv6 dhcp interface</pre>	Verifies that the DHCPv6 client is enabled on an interface.

Displaying IPv6

For complete syntax and usage information on these commands, see the Cisco IOS command reference publications.

Table 4: Command for Monitoring IPv6

Command	Purpose
show ipv6 access-list	Displays a summary of access lists.
show ipv6 cef	Displays Cisco Express Forwarding for IPv6.
show ipv6 interface <i>interface-id</i>	Displays IPv6 interface status and configuration.
show ipv6 mtu	Displays IPv6 MTU per destination cache.
show ipv6 neighbors	Displays IPv6 neighbor cache entries.
show ipv6 prefix-list	Displays a list of IPv6 prefix lists.
show ipv6 protocols	Displays a list of IPv6 routing protocols on the switch.
show ipv6 rip	Displays IPv6 RIP routing protocol status.
show ipv6 route	Displays IPv6 route table entries.
show ipv6 static	Displays IPv6 static routes.
show ipv6 traffic	Displays IPv6 traffic statistics.

Configuration Examples for IPv6 Unicast Routing

Configuring IPv6 Addressing and Enabling IPv6 Routing: Example

This example shows how to enable IPv6 with both a link-local address and a global address based on the IPv6 prefix 2001:0DB8: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 (20B:46FF:FE2F:D940) is appended to the link-local prefix FE80::/64 of the interface.

```
(config)# ipv6 unicast-routing
(config)# interface gigabitethernet0/11

(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
(config-if)# end
# show ipv6 interface gigabitethernet0/11
GigabitEthernet0/11 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
Global unicast address(es):
2001:0DB8:c18:1:20B:46FF:FE2F:D940, subnet is 2001:0DB8:c18:1::/64 [EUI]
Joined group address(es):
```

```

FF02::1
FF02::2
FF02::1:FF2F:D940
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.

```

Configuring Default Router Preference: Example

This example shows how to configure a DRP of *high* for the router on an interface.

```

# configure terminal
(config)# interface gigabitethernet1/0/1
(config-if)# ipv6 nd router-preference high
(config-if)# end

```

Configuring IPv4 and IPv6 Protocol Stacks: Example

This example shows how to enable IPv4 and IPv6 routing on an interface.

```

(config)# ip routing
(config)# ipv6 unicast-routing
(config)# interface fastethernet1/0/11
(config-if)# no switchport
(config-if)# ip address 192.168.99.1 255.255.255.0
(config-if)# ipv6 address 2001:0DB8:c18:1::/64 eui 64
(config-if)# end

```

Enabling DHCPv6 Server Function: Example

This example shows how to configure a pool called *engineering* with an IPv6 address prefix:

```

# configure terminal
(config)# ipv6 dhcp pool engineering
(config-dhcpv6)#address prefix 2001:1000::0/64
(config-dhcpv6)# end

```

This example shows how to configure a pool called *testgroup* with three link-addresses and an IPv6 address prefix:

```

# configure terminal
(config)# ipv6 dhcp pool testgroup
(config-dhcpv6)# link-address 2001:1001::0/64
(config-dhcpv6)# link-address 2001:1002::0/64

```

```
(config-dhcpv6) # link-address 2001:2000::0/48
(config-dhcpv6) # address prefix 2001:1003::0/64
(config-dhcpv6) # end
```

This example shows how to configure a pool called *350* with vendor-specific options:

```
# configure terminal
(config) # ipv6 dhcp pool 350
(config-dhcpv6) # address prefix 2001:1005::0/48
(config-dhcpv6) # vendor-specific 9
(config-dhcpv6-vs) # suboption 1 address 1000:235D::1
(config-dhcpv6-vs) # suboption 2 ascii "IP-Phone"
(config-dhcpv6-vs) # end
```

Enabling DHCPv6 Client Function: Example

This example shows how to acquire an IPv6 address and to enable the rapid-commit option:

```
(config) # interface gigabitethernet2/0/1
(config-if) # ipv6 address dhcp rapid-commit
```

Configuring IPv6 ICMP Rate Limiting: Example

This example shows how to configure an IPv6 ICMP error message interval of 50 milliseconds and a bucket size of 20 tokens.

```
(config) # ipv6 icmp error-interval 50 20
```

Configuring Static Routing for IPv6: Example

This example shows how to configure a floating static route to an interface with an administrative distance of 130:

```
(config) # ipv6 route 2001:0DB8::/32 gigabitethernet 1/0/1 130
```

Example: Enabling PBR on an Interface

In the following example, a route map named *pbr-dest-1* is created and configured, specifying packet match criteria and desired policy-route action. PBR is then enabled on GigabitEthernet interface *0/0/1*.

```
ipv6 access-list match-dest-1
 permit ipv6 any 2001:DB8:2001:1760::/32
route-map pbr-dest-1 permit 10
 match ipv6 address match-dest-1
 set interface GigabitEthernet 0/0/0
interface GigabitEthernet0/0/1
 ipv6 policy-route-map interactive
```

Example: Enabling Local PBR for IPv6

In the following example, packets with a destination IPv6 address that match the IPv6 address range allowed by access list pbr-src-90 are sent to the device at IPv6 address 2001:DB8:2003:1::95:

```
ipv6 access-list src-90
  permit ipv6 host 2001:DB8:2003::90 2001:DB8:2001:1000::/64
route-map pbr-src-90 permit 10
  match ipv6 address src-90
  set ipv6 next-hop 2001:DB8:2003:1::95
ipv6 local policy route-map pbr-src-90
```

Configuring RIP for IPv6: Example

This example shows how to enable the RIP routing process *cisco* with a maximum of eight equal-cost routes and to enable it on an interface:

```
(config)# ipv6 router rip cisco
(config-router)# maximum-paths 8
(config)# exit
(config)# interface gigabitethernet2/0/11
(config-if)# ipv6 rip cisco enable
```

Displaying IPv6: Example

This is an example of the output from the **show ipv6 interface** privileged EXEC command:

```
# show ipv6 interface
Vlan1 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::20B:46FF:FE2F:D940
Global unicast address(es):
  3FFE:C000:0:1:20B:46FF:FE2F:D940, subnet is 3FFE:C000:0:1::/64 [EUI]
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF2F:D940
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
<output truncated>
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Standards and RFCs

Standard/RFC	Title
RFC 5453	<i>Reserved IPv6 Interface Identifiers</i>

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs



CHAPTER 4

Configuring RIP

- [Information About RIP, on page 57](#)
- [How to Configure RIP, on page 58](#)
- [Example: Configuring RIP for IPv6, on page 67](#)
- [Configuration Example for Summary Addresses and Split Horizon , on page 67](#)

Information About RIP

The Routing Information Protocol (RIP) is an interior gateway protocol (IGP) created for use in small, homogeneous networks. It is a distance-vector routing protocol that uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information. The protocol is documented in RFC 1058. You can find detailed information about RIP in *IP Routing Fundamentals*, published by Cisco Press.

Using RIP, the switch sends routing information updates (advertisements) every 30 seconds. If a router does not receive an update from another router for 180 seconds or more, it marks the routes served by that router as unusable. If there is still no update after 240 seconds, the router removes all routing table entries for the non-updating router.

RIP uses hop counts to rate the value of different routes. The hop count is the number of routers that can be traversed in a route. A directly connected network has a hop count of zero; a network with a hop count of 16 is unreachable. This small range (0 to 15) makes RIP unsuitable for large networks.

If the router has a default network path, RIP advertises a route that links the router to the pseudonetwork 0.0.0.0. The 0.0.0.0 network does not exist; it is treated by RIP as a network to implement the default routing feature. The switch advertises the default network if a default was learned by RIP or if the router has a gateway of last resort and RIP is configured with a default metric. RIP sends updates to the interfaces in specified networks. If an interface's network is not specified, it is not advertised in any RIP update.



Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

RIP for IPv6

Routing Information Protocol (RIP) for IPv6 is a distance-vector protocol that uses hop count as a routing metric. It includes support for IPv6 addresses and prefixes and the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages.

For configuring RIP for IPv6, see the *Configuring RIP for IPv6* section.

For more information about RIP for IPv6, see the “Implementing RIP for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Summary Addresses and Split Horizon

Routers connected to broadcast-type IP networks and using distance-vector routing protocols normally use the split-horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router on any interface from which that information originated. This feature usually optimizes communication among multiple routers, especially when links are broken.

How to Configure RIP

Default RIP Configuration

Table 5: Default RIP Configuration

Feature	Default Setting
Auto summary	Enabled.
Default-information originate	Disabled.
Default metric	Built-in; automatic metric translations.
IP RIP authentication key-chain	No authentication. Authentication mode: clear text.
IP RIP triggered	Disabled
IP split horizon	Varies with media.
Neighbor	None defined.
Network	None specified.
Offset list	Disabled.
Output delay	0 milliseconds.

Feature	Default Setting
Timers basic	<ul style="list-style-type: none"> • Update: 30 seconds. • Invalid: 180 seconds. • Hold-down: 180 seconds. • Flush: 240 seconds.
Validate-update-source	Enabled.
Version	Receives RIP Version 1 and 2 packets; sends Version 1 packets.

Configuring Basic RIP Parameters

To configure RIP, you enable RIP routing for a network and optionally configure other parameters. On the switch, RIP configuration commands are ignored until you configure the network number.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip routing Example: Device(config)# ip routing	Enables IP routing. (Required only if IP routing is disabled.)
Step 4	router rip Example: Device(config)# router rip	Enables a RIP routing process, and enter router configuration mode.
Step 5	network <i>network number</i> Example: Device(config-router)# network 12.0.0.0	Associates a network with a RIP routing process. You can specify multiple network commands. RIP routing updates are sent and received through interfaces only on these networks.

	Command or Action	Purpose
		Note You must configure a network number for the RIP commands to take effect.
Step 6	neighbor <i>ip-address</i> Example: Device(config-router)# neighbor 10.2.5.1	(Optional) Defines a neighboring router with which to exchange routing information. This step allows routing updates from RIP (normally a broadcast protocol) to reach nonbroadcast networks.
Step 7	offset-list [<i>access-list number name</i>] { in out } <i>offset</i> [<i>type number</i>] Example: Device(config-router)# offset-list 103 in 10	(Optional) Applies an offset list to routing metrics to increase incoming and outgoing metrics to routes learned through RIP. You can limit the offset list with an access list or an interface.
Step 8	timers basic <i>update invalid holddown flush</i> Example: Device(config-router)# timers basic 45 360 400 300	(Optional) Adjusts routing protocol timers. Valid ranges for all timers are 0 to 4294967295 seconds. <ul style="list-style-type: none"> • <i>update</i>—The time between sending routing updates. The default is 30 seconds. • <i>invalid</i>—The timer after which a route is declared invalid. The default is 180 seconds. • <i>holddown</i>—The time before a route is removed from the routing table. The default is 180 seconds. • <i>flush</i>—The amount of time for which routing updates are postponed. The default is 240 seconds.
Step 9	version { 1 2 } Example: Device(config-router)# version 2	(Optional) Configures the switch to receive and send only RIP Version 1 or RIP Version 2 packets. By default, the switch receives Version 1 and 2 but sends only Version 1. You can also use the interface commands ip rip {send receive} version 1 2 1 2 to control what versions are used for sending and receiving on interfaces.
Step 10	no auto summary Example: Device(config-router)# no auto summary	(Optional) Disables automatic summarization. By default, the switch summarizes subprefixes when crossing classful network boundaries. Disable summarization (RIP Version 2 only) to advertise subnet and host routing information to classful network boundaries.
Step 11	output-delay <i>delay</i> Example: Device(config-router)# output-delay 8	(Optional) Adds interpacket delay for RIP updates sent. By default, packets in a multiple-packet RIP update have no delay added between packets. If you are sending packets to a lower-speed device, you can add an interpacket delay in the range of 8 to 50 milliseconds.
Step 12	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config-router)# end	
Step 13	show ip protocols Example: Device# show ip protocols	Verifies your entries.
Step 14	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring RIP Authentication

RIP Version 1 does not support authentication. If you are sending and receiving RIP Version 2 packets, you can enable RIP authentication on an interface. The key chain specifies the set of keys that can be used on the interface. If a key chain is not configured, no authentication is performed, not even the default.

The switch supports two modes of authentication on interfaces for which RIP authentication is enabled: plain text and MD5. The default is plain text.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the interface to configure.
Step 4	ip rip authentication key-chain <i>name-of-chain</i> Example: Device(config-if)# ip rip authentication key-chain trees	Enables RIP authentication.

	Command or Action	Purpose
Step 5	ip rip authentication mode {text md5} Example: Device(config-if)# ip rip authentication mode md5	Configures the interface to use plain text authentication (the default) or MD5 digest authentication.
Step 6	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 7	show running-config Example: Device# show running-config	Verifies your entries.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring RIP for IPv6

For more information about configuring RIP routing for IPv6, see the “Implementing RIP for IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com,

To configure RIP routing for IPv6, perform this procedure:

Before you begin

Before configuring the switch to run IPv6 RIP, you must enable routing by using the **ip routing** command in global configuration mode, enable the forwarding of IPv6 packets by using the **ipv6 unicast-routing** command in global configuration mode, and enable IPv6 on any Layer 3 interfaces on which IPv6 RIP is to be enabled.

Procedure

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

	Command or Action	Purpose
Step 3	ipv6 router rip <i>name</i> Example: Device(config)# ipv6 router rip cisco	Configures an IPv6 RIP routing process, and enters router configuration mode for the process.
Step 4	maximum-paths <i>number-paths</i> Example: Device(config-router)# maximum-paths 6	(Optional) Define the maximum number of equal-cost routes that IPv6 RIP can support. The range is from 1 to 32, and the default is 16 routes.
Step 5	exit Example: Device(config-router)# exit	Returns to global configuration mode.
Step 6	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 7	ipv6 rip <i>name</i> enable Example: Device(config-if)# ipv6 rip cisco enable	Enables the specified IPv6 RIP routing process on the interface.
Step 8	ipv6 rip <i>name</i> default-information { only originate } Example: Device(config-if)# ipv6 rip cisco default-information only	(Optional) Originates the IPv6 default route (::/0) into the RIP routing process updates sent from the specified interface. Note To avoid routing loops after the IPv6 default route (::/0) is originated from any interface, the routing process ignores all default routes received on any interface. <ul style="list-style-type: none"> • only—Select to originate the default route, but suppress all other routes in the updates sent on this interface. • originate—Select to originate the default route in addition to all other routes in the updates sent on this interface.
Step 9	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 10	Use one of the following: <ul style="list-style-type: none"> • show ipv6 rip [<i>name</i>] [interface <i>interface-id</i>] [database] [next-hops] • show ipv6 rip Example:	<ul style="list-style-type: none"> • Displays information about current IPv6 RIP processes. • Displays the current contents of the IPv6 routing table.

	Command or Action	Purpose
	<pre>Device# show ipv6 rip cisco interface gigabitethernet 2/0/1 or Device# show ipv6 rip</pre>	
Step 11	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Summary Addresses and Split Horizon



Note In general, disabling split horizon is not recommended unless you are certain that your application requires it to properly advertise routes.

If you want to configure an interface running RIP to advertise a summarized local IP address pool on a network access server for dial-up clients, use the **ip summary-address rip** interface configuration command.



Note If split horizon is enabled, neither autosummary nor interface IP summary addresses are advertised.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	Enters interface configuration mode, and specifies the Layer 3 interface to configure.
Step 4	ip address <i>ip-address subnet-mask</i> Example:	Configures the IP address and IP subnet.

	Command or Action	Purpose
	Device(config-if)# ip address 10.1.1.10 255.255.255.0	
Step 5	ip summary-address rip ip address ip-network mask Example: Device(config-if)# ip summary-address rip ip address 10.1.1.30 255.255.255.0	Configures the IP address to be summarized and the IP network mask.
Step 6	no ip split horizon Example: Device(config-if)# no ip split horizon	Disables split horizon on the interface.
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 8	show ip interface interface-id Example: Device# show ip interface gigabitethernet 1/0/1	Verifies your entries.
Step 9	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Split Horizon

Routers connected to broadcast-type IP networks and using distance-vector routing protocols normally use the split-horizon mechanism to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router on any interface from which that information originated. This feature can optimize communication among multiple routers, especially when links are broken.



Note In general, we do not recommend disabling split horizon unless you are certain that your application requires it to properly advertise routes.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Enters interface configuration mode, and specifies the interface to configure.
Step 4	ip address <i>ip-address subnet-mask</i> Example: Device(config-if)# ip address 10.1.1.10 255.255.255.0	Configures the IP address and IP subnet.
Step 5	no ip split-horizon Example: Device(config-if)# no ip split-horizon	Disables split horizon on the interface.
Step 6	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 7	show ip interface <i>interface-id</i> Example: Device# show ip interface gigabitethernet 1/0/1	Verifies your entries.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example: Configuring RIP for IPv6

This example shows how to enable the RIP routing process *cisco* with a maximum of eight equal-cost routes and to enable it on an interface:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 router rip cisco
Device(config-router)# maximum-paths 8
Device(config)# exit
Device(config)# interface gigabitethernet2/0/11
Device(config-if)# ipv6 rip cisco enable
```

Configuration Example for Summary Addresses and Split Horizon

In this example, the major net is 10.0.0.0. The summary address 10.2.0.0 overrides the autosummary address of 10.0.0.0 so that 10.2.0.0 is advertised out interface Gigabit Ethernet port 2, and 10.0.0.0 is not advertised. In the example, if the interface is still in Layer 2 mode (the default), you must enter a **no switchport** interface configuration command before entering the **ip address** interface configuration command.



Note If split horizon is enabled, neither autosummary nor interface summary addresses (those configured with the **ip summary-address rip** router configuration command) are advertised.

```
Device(config)# router rip
Device(config-router)# interface gigabitethernet1/0/2
Device(config-if)# ip address 10.1.5.1 255.255.255.0
Device(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0
Device(config-if)# no ip split-horizon
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# network 10.0.0.0
Device(config-router)# neighbor 2.2.2.2 peer-group mygroup
Device(config-router)# end
```




CHAPTER 5

Configuring VRF-lite

- [Information About VRF-lite, on page 69](#)
- [Guidelines for Configuring VRF-lite, on page 71](#)
- [How to Configure VRF-lite, on page 72](#)
- [Configuring VRF-lite for IPv6, on page 78](#)
- [Additional Information for VRF-lite, on page 90](#)
- [Verifying VRF-lite Configuration, on page 91](#)
- [Configuration Examples for VRF-lite, on page 93](#)

Information About VRF-lite

VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but a Layer 3 interface cannot belong to more than one VRF at any time.



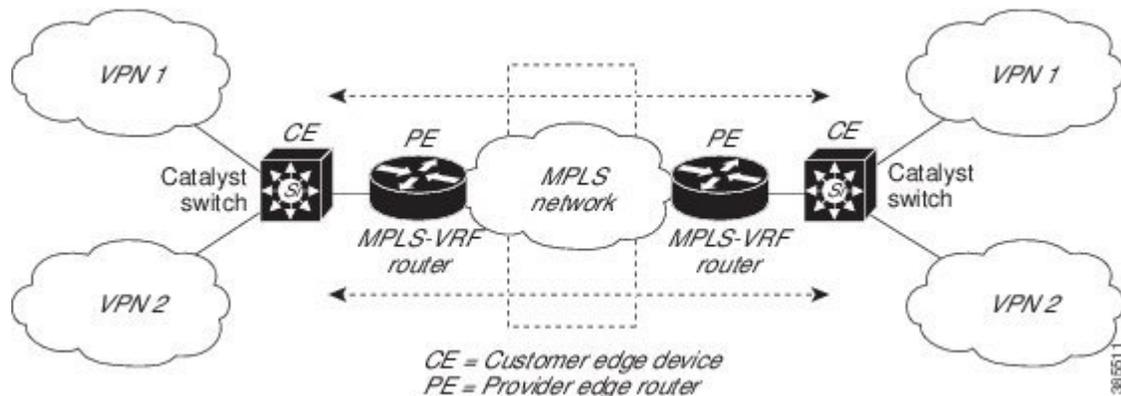
Note VRF-lite interfaces must be Layer 3 interfaces.

VRF-lite includes these devices:

- Customer edge (CE) devices provide customer access to the service provider network over a data link to one or more provider edge routers. The CE device advertises the site's local routes to the provider edge router and learns the remote VPN routes from it. A Cisco Catalyst Switch can be a CE.
- Provider routers (or core routers) are any routers in the service provider network that do not attach to CE devices.

The following figure displays a configuration where each Cisco Catalyst switch acts as multiple virtual CEs. Because VRF-lite is a Layer 3 feature, each interface in a VRF must be a Layer 3 interface.

Figure 4: Cisco Catalyst Switches Acting as Multiple Virtual CEs



This figure illustrates the packet-forwarding process in a VRF-lite CE-enabled network.

- When the CE receives a packet from a VPN, it looks up the routing table based on the input interface. When a route is found, the CE forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it performs a VRF lookup. When a route is found, the router adds a corresponding MPLS label to the packet and sends it to the MPLS network.
- When an egress PE receives a packet from the network, it strips the label and uses the label to identify the correct VPN routing table. The egress PE then performs the normal route lookup. When a route is found, it forwards the packet to the correct adjacency.
- When a CE receives a packet from an egress PE, it uses the input interface to look up the correct VPN routing table. If a route is found, the CE forwards the packet within the VPN.

To configure VRF, create a VRF table and specify the Layer 3 interface associated with the VRF. You then configure the routing protocols in the VPN and between the CE and the PE. BGP is the preferred routing protocol used to distribute VPN routing information across the providers' backbone. The VRF-lite network has three major components:

- VPN route target communities—Lists all other members of a VPN community. You need to configure VPN route targets for each VPN community member.
- Multiprotocol BGP peering of VPN community PE routers—Propagates VRF reachability information to all members of a VPN community. You need to configure BGP peering in all PE routers within a VPN community.
- VPN forwarding—Transports all traffic between all VPN community members across a VPN service-provider network.



Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Guidelines for Configuring VRF-lite

IPv4 and IPv6

- A switch with VRF-lite is shared by multiple customers, and all customers have their own routing tables.
- Because customers use different VRF tables, you can reuse the same IP addresses. Overlapped IP addresses are allowed in different VPNs.
- VRF-lite lets multiple customers share the same physical link between the PE and the CE. Trunk ports with multiple VLANs separate packets among customers. All customers have their own VLANs.
- For the PE router, there is no difference between using VRF-lite or using multiple CEs. In [#unique_104](#), multiple virtual Layer 3 interfaces are connected to the VRF-lite device.
- The Cisco Catalyst switch supports configuring VRF by using physical ports, VLAN SVIs, or a combination of both. You can connect SVIs through an access port or a trunk port.
- A customer can use multiple VLANs as long as they do not overlap with those of other customers. A customer's VLANs are mapped to a specific routing table ID that is used to identify the appropriate routing tables stored on the switch.
- The Layer 3 TCAM resource is shared between all VRFs. To ensure that any one VRF has sufficient CAM space, use the **maximum routes** command.
- A Cisco Catalyst switch using VRF can support one global network and multiple VRFs. The total number of routes supported is limited by the size of the TCAM.
- A single VRF can be configured for both IPv4 and IPv6.
- If an incoming packet's destination address is not found in the vrf table, the packet is dropped. Also, if insufficient TCAM space exists for a VRF route, hardware switching for that VRF is disabled and the corresponding data packets are sent to software for processing.

IPv4 Specific

- You can use most routing protocols (BGP, OSPF, EIGRP, RIP and static routing) between the CE and the PE. However, we recommend using external BGP (EBGP) for these reasons:
 - BGP does not require multiple algorithms to communicate with multiple CEs.
 - BGP is designed for passing routing information between systems run by different administrations.
 - BGP makes simplifies passing attributes of the routes to the CE.
- Multicast VRF-lite is not supported.
- The **capability vrf-lite** subcommand under **router ospf** should be used when configuring OSPF as the routing protocol between the PE and the CE.

IPv6 specific

- VRF-aware OSPFv3, BGPv6, EIGRPv6, and IPv6 static routing are supported.

- VRF-aware IPv6 route applications include: ping, telnet, ssh, tftp, ftp and traceroute. (This list does not include the Mgt interface, which is handled differently even though you can configure both IPv4 or IPv6 VRF under it.)

How to Configure VRF-lite

Configuring VRF-lite for IPv4

Configuring VRF-Aware Services

IP services can be configured on global interfaces and within the global routing instance. IP services are enhanced to run on multiple routing instances; they are VRF-aware. Any configured VRF in the system can be specified for a VRF-aware service.

VRF-aware services are implemented in platform-independent modules. VRF provides multiple routing instances in Cisco IOS. Each platform has its own limit on the number of VRFs it supports.

VRF-aware services have the following characteristics:

- The user can ping a host in a user-specified VRF.
- ARP entries are learned in separate VRFs. The user can display Address Resolution Protocol (ARP) entries for specific VRFs.

Configuring the User Interface for ARP

Procedure

	Command or Action	Purpose
Step 1	show ip arp vrf vrf-name Example: Switch# show ip arp vrf vrf-name	Displays the ARP table (static and dynamic entries) in the specified VRF.
Step 2	arp vrf vrf-name ip-address mac-address ARPA Example: Switch(config)# arp vrf vrf-name ip-address mac-address ARPA	Creates a static ARP entry in the specified VRF.

Configuring Per-VRF for TACACS+ Servers

The per-VRF for TACACS+ servers feature enables you to configure per-virtual route forwarding (per-VRF) authentication, authorization, and accounting (AAA) on TACACS+ servers.

You can create the VRF routing table (shown in Steps 3 and 4) and configure the interface (Steps 6, 7, and 8). The actual configuration of per-VRF on a TACACS+ server is done in Steps 10 through 13.

Before you begin

Before configuring per-VRF on a TACACS+ server, you must have configured AAA and a server group.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 3	vrf definition <i>vrf-name</i> Example: Switch(config)# ip vrf vrf-name	Names the VRF and enters VRF configuration mode.
Step 4	rd <i>route-distinguisher</i> Example: Switch (config-vrf)# rd route-distinguisher	Creates routing and forwarding tables for a VRF instance.
Step 5	exit Example: Switch (config-vrf)# exit	Exits VRF configuration mode.
Step 6	interface <i>interface-name</i> Example: Switch (config)# interface interface-name	Configures an interface and enters interface configuration mode.
Step 7	vrf forwarding <i>vrf-name</i> Example: Switch (config-if)# vrf forwarding vrf-name	Configures a VRF for the interface.
Step 8	ip address <i>ip-address mask [secondary]</i> Example: Switch (config-if)# ip address ip-address mask [secondary]	Sets a primary or secondary IP address for an interface.
Step 9	exit Example: Switch (config-vrf)# exit	Exits interface configuration mode.
Step 10	aaa group server tacacs+ <i>group-name</i> Example: Switch (config)# aaa group server tacacs+ tacacs1	Groups different TACACS+ server hosts into distinct lists and distinct methods and enters server-group configuration mode.

	Command or Action	Purpose
Step 11	server-private <i>{ip-address name}</i> [nat] [single-connection] [port <i>port-number</i>] [timeout <i>seconds</i>] [key [0 7] <i>string</i>] Example: Switch (config-sg-tacacs)# server-private 10.1.1.1 port 19 key cisco	Configures the IP address of the private TACACS+ server for the group server.
Step 12	vrf forwarding <i>vrf-name</i> Example: Switch (config-sg-tacacs)# vrf forwarding vrf-name	Configures the VRF reference of a AAA TACACS+ server group.
Step 13	ip tacacs source-interface <i>subinterface-name</i> Example: Switch (config-sg-tacacs)# ip tacacs source-interface subinterface-name	Uses the IP address of a specified interface for all outgoing TACACS+ packets.
Step 14	exit Example: Switch (config-sg-tacacs)# exit	Exits server-group configuration mode.

Example

The following example lists all the steps to configure per-VRF TACACS+:

```
Switch> enable
Switch# configure terminal
Switch (config)# vrf definition cisco
Switch (config-vrf)# rd 100:1
Switch (config-vrf)# exit
Switch (config)# interface Loopback0
Switch (config-if)# vrf forwarding cisco
Switch (config-if)# ip address 10.0.0.2 255.0.0.0
Switch (config-if)# exit
Switch (config-sg-tacacs)# vrf forwarding cisco
Switch (config-sg-tacacs)# ip tacacs source-interface Loopback0
Switch (config-sg-tacacs)# exit
```

Configuring a VPN Routing Session

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 2	router ospf <i>process-id</i> vrf <i>vrf-name</i> Example: Switch(config)# router ospf process-id vrf vrf-name	Enables OSPF routing, specifies a VPN forwarding table, and enters router configuration mode.
Step 3	capability vrf-lite Example: Switch(config-router)# capability vrf-lite	
Step 4	log-adjacency-changes Example: Switch(config-router)# log-adjacency-changes	(Optional) Logs changes in the adjacency state (the default state).
Step 5	redistribute bgp <i>autonomous-system-number</i> <i>subnets</i> Example: Switch(config-router)# redistribute bgp autonomous-system-number subnets	Sets the switch to redistribute information from the BGP network to the OSPF network.
Step 6	network <i>network-number</i> area <i>area-id</i> Example: Switch(config-router)# network network-number area area-id	Defines a network address and mask on which OSPF runs and the area ID for that network address.
Step 7	end Example: Switch(config-router)# end	Returns to privileged EXEC mode.
Step 8	show ip ospf <i>process-id</i> Example: Switch# show ip ospf process-id	Verifies the configuration of the OSPF network.
Step 9	copy running-config startup-config Example: Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file. Use the no router ospf process-id vrf vrf-name global configuration command to disassociate the VPN forwarding table from the OSPF routing process.

Example

```
Switch(config)# vrf definition VRF-RED
Switch(config-vrf)# rd 1:1
Switch(config-vrf)# exit
Switch(config)# router eigrp virtual-name
Switch(config-router)# address-family ipv4 vrf VRF-RED autonomous-system 1
Switch(config-router-af)# network 10.0.0.0 0.0.0.255
Switch(config-router-af)# topology base
Switch(config-router-topology)# default-metric 10000 100 255 1 1500
Switch(config-router-topology)# exit-af-topology
Switch(config-router-af)# exit-address-family
```

Configuring BGP PE to CE Routing Sessions

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: Switch(config)# router bgp autonomous-system-number	Configures the BGP routing process with the AS number passed to other BGP routers and enters router configuration mode.
Step 3	network <i>network-number</i> mask <i>network-mask</i> Example: Switch(config-router)# network network-number mask network-mask	Specifies a network and mask to announce using BGP.
Step 4	redistribute ospf <i>process-id</i> match <i>internal</i> Example: Switch(config-router)# redistribute ospf process-id match internal	Sets the switch to redistribute OSPF internal routes.
Step 5	network <i>network-number</i> area <i>area-id</i> Example: Switch(config-router)# network network-number area area-id	Defines a network address and mask on which OSPF runs and the area ID for that network address.
Step 6	address-family ipv4 vrf <i>vrf-name</i> Example: Switch(config-router-af)# address-family ipv4 vrf vrf-name	Defines BGP parameters for PE to CE routing sessions and enters VRF address-family mode.
Step 7	neighbor <i>address</i> remote-as <i>as-number</i> Example: Switch(config-router-af)# neighbor address remote-as as-number	Defines a BGP session between PE and CE routers.
Step 8	neighbor <i>address</i> activate Example: Switch(config-router-af)# neighbor address activate	Activates the advertisement of the IPv4 address family.
Step 9	end Example: Switch(config-router-af)# end	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 10	show ip bgp [ipv4] [neighbors] Example: Switch# show ip bgp [ipv4] [neighbors]	Verifies BGP configuration. Use the no router bgp autonomous-system-number global configuration command to delete the BGP routing process. Use the command with keywords to delete routing characteristics.

Configuring IPv4 VRFs

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	ip routing Example: Switch# configure terminal	Enters global configuration mode.
Step 3	vrf definition vrf-name Example: Switch(config)# vrf definition vrf-name	Names the VRF and enters VRF configuration mode.
Step 4	rd route-distinguisher Example: Switch(config-vrf)# rd route-distinguisher	Creates a VRF table by specifying a route distinguisher. Enter either an Autonomous System number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 5	route-target {export import both} route-target-ext-community Example: Switch(config-vrf)# route-target {export import both} route-target-ext-community	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). Note This command is effective only if BGP is running.
Step 6	import map route-map Example: Switch(config-vrf)# import map route-map	(Optional) Associates a route map with the VRF.
Step 7	interface interface-id Example: Switch(config-vrf)# interface interface-id	Enters interface configuration mode and specify the Layer 3 interface to be associated with the VRF. The interface can be a routed port or SVI.

	Command or Action	Purpose
Step 8	vrf forwarding <i>vrf-name</i> Example: Switch(config-if)# vrf forwarding vrf-name	Associates the VRF with the Layer 3 interface.
Step 9	end Example: Switch(config-if)# end	Returns to privileged EXEC mode.
Step 10	show ip vrf [brief detail interfaces] [<i>vrf-name</i>] Example: Switch# show ip vrf [brief detail interfaces] [<i>vrf-name</i>]	Verifies the configuration. Displays information about the configured VRFs.
Step 11	copy running-config startup-config Example: Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file. Note For complete syntax and usage information for the following commands, see the switch command reference for this release and see the Cisco IOS Switching Services Command Reference . Use the no vrf definition <i>vrf-name</i> global configuration command to delete a VRF and to remove all interfaces from it. Use the no vrf forwarding interface configuration command to remove an interface from the VRF.

Configuring VRF-lite for IPv6

Configuring VRF-Aware Services

IPv6 services can be configured on global interfaces and within the global routing instance. IPv6 services are enhanced to run on multiple routing instances; they are VRF-aware. Any configured VRF in the system can be specified for a VRF-aware service.

VRF-aware services are implemented in platform-independent modules. VRF provides multiple routing instances in Cisco IOS. Each platform has its own limit on the number of VRFs it supports.

VRF-aware services have the following characteristics:

- The user can ping a host in a user-specified VRF.
- Neighbor Discovery entries are learned in separate VRFs. The user can display Neighbor Discovery (ND) entries for specific VRFs.

The following services are VRF-aware:

- Ping
- Unicast Reverse Path Forwarding (uRPF)

- Traceroute
- FTP and TFTP
- Telnet and SSH
- NTP

Configuring the User Interface for PING

Perform the following task to configure a VRF-aware ping:

Procedure

	Command or Action	Purpose
Step 1	<p>ping vrf <i>vrf-name</i> ipv6-host</p> <p>Example:</p> <pre>Switch# ping vrf vrf-name ipv6-host</pre>	Pings an IPv6 host or address in the specified VRF.

Configuring the User Interface for uRPF

You can configure uRPF on an interface assigned to a VRF. Source lookup is performed in the VRF table

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-id*
3. **no switchport**
4. **vrf forwarding** *vrf-name*
5. **ipv6 address** *ip-address* *subnet-mask*
6. **ipv6 verify unicast source reachable-via rx allow-default**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>configure terminal</p> <p>Example:</p> <pre>Switch# configure terminal</pre>	Enters global configuration mode.
Step 2	<p>interface <i>interface-id</i></p> <p>Example:</p> <pre>Switch (config)# interface interface-id</pre>	Enters interface configuration mode and specifies the Layer 3 interface to configure.
Step 3	<p>no switchport</p> <p>Example:</p> <pre>Switch (config-if)# no switchport</pre>	Removes the interface from Layer 2 configuration mode if it is a physical interface.

	Command or Action	Purpose
Step 4	vrf forwarding <i>vrf-name</i> Example: Switch (config-if)# vrf forwarding vrf-name	Configures VRF on the interface.
Step 5	ipv6 address <i>ip-address subnet-mask</i> Example: Switch (config-if)# ip address ip-address mask	Enters the IPv6 address for the interface.
Step 6	ipv6 verify unicast source reachable-via rx allow-default Example: Switch(config-if)# ipv6 verify unicast source reachable-via rx allow-default	Enables uRPF on the interface.
Step 7	end Example: Switch(config-if)# end	Returns to privileged EXEC mode.

Configuring the User Interface for Traceroute

SUMMARY STEPS

1. **traceroute vrf** *vrf-name ipv6address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	traceroute vrf <i>vrf-name ipv6address</i> Example: Switch# traceroute vrf vrf-name ipv6address	Specifies the name of a VPN VRF in which to find the destination address.

Configuring the User Interface for Telnet and SSH

SUMMARY STEPS

1. **telnet** *ipv6-address/vrf vrf-name*
2. **ssh -l** *username -vrf vrf-name ipv6-host*

DETAILED STEPS

	Command or Action	Purpose
Step 1	telnet <i>ipv6-address/vrf vrf-name</i> Example: Switch# telnet ipv6-address/vrf vrf-name	Connects through Telnet to an IPv6 host or address in the specified VRF.

	Command or Action	Purpose
Step 2	ssh -l <i>username</i> -vrf <i>vrf-name</i> ipv6-host Example: Switch# ssh -l username -vrf vrf-name ipv6-host	Connects through SSH to an IPv6 host or address in the specified VRF.

Configuring the User Interface for NTP

SUMMARY STEPS

1. **configure terminal**
2. **ntp server vrf *vrf-name* ipv6-host**
3. **ntp peer vrf *vrf-name* ipv6-host**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: # configure terminal	Enters global configuration mode.
Step 2	ntp server vrf <i>vrf-name</i> ipv6-host Example: (config)# ntp server vrf vrf-name ipv6-host	Configure the NTP server in the specified VRF.
Step 3	ntp peer vrf <i>vrf-name</i> ipv6-host Example: (config)# ntp peer vrf vrf-name ipv6-host	Configure the NTP peer in the specified VRF.

Configuring IPv6 VRFs

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition *vrf-name***
3. **rd *route-distinguisher***
4. **address-family *ipv4* | *ipv6***
5. **route-target {*export* | *import* | *both*} *route-target-ext-community***
6. **exit-address-family**
7. **vrf definition *vrf-name***
8. **ipv6 multicast multitopology**
9. **address-family ipv6 multicast**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	vrf definition vrf-name Example: Switch(config)# vrf definition vrf-name	Names the VRF and enters VRF configuration mode.
Step 3	rd route-distinguisher Example: Switch(config-vrf)# rd route-distinguisher	(Optional) Creates a VRF table by specifying a route distinguisher. Enter either an Autonomous System number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 4	address-family ipv4 ipv6 Example: Switch(config-vrf)# address-family ipv4 ipv6	(Optional) IPv4 by default. Configuration MUST for IPv6.
Step 5	route-target {export import both} <i>route-target-ext-community</i> Example: Switch(config-vrf)# route-target {export import both} route-target-ext-community	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). Note This command is effective only if BGP is running.
Step 6	exit-address-family Example: Switch(config-vrf)# exit-address-family	Exits VRF address-family configuration mode and return to VRF configuration mode.
Step 7	vrf definition vrf-name Example: Switch(config)# vrf definition vrf-name	Enters VRF configuration mode.
Step 8	ipv6 multicast multitopology Example: Switch(config-vrf-af)# ipv6 multicast multitopology	Enables multicast specific RPF topology.
Step 9	address-family ipv6 multicast Example: Switch(config-vrf)# address-family ipv6 multicast	Enter multicast IPv6 address-family.
Step 10	end Example: Switch(config-vrf-af)# end	Returns to privileged EXEC mode.

Example

This example shows how to configure VRFs:

```
Switch(config)# vrf definition red
Switch(config-vrf)# rd 100:1
Switch(config-vrf)# address family ipv6
Switch(config-vrf-af)# route-target both 200:1
Switch(config-vrf)# exit-address-family
Switch(config-vrf)# vrf definition red
Switch(config-if)# ipv6 multicast multitopology
Switch(config-if)# address-family ipv6 multicast
Switch(config-vrf-af)# end
Switch#
```

Associating Interfaces to the Defined VRFs

SUMMARY STEPS

1. **interface** *interface-id*
2. **no switchport**
3. **vrf forwarding** *vrf-name*
4. **ipv6 enable**
5. **ipv6 address** *ip-address subnet-mask*
6. **show ipv6 vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*]
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	interface <i>interface-id</i> Example: Switch(config-vrf)# interface interface-id	Enters interface configuration mode and specify the Layer 3 interface to be associated with the VRF. The interface can be a routed port or SVI.
Step 2	no switchport Example: Switch(config-if)# no switchport	Removes the interface from configuration mode if it is a physical interface.
Step 3	vrf forwarding <i>vrf-name</i> Example: Switch(config-if)# vrf forwarding vrf-name	Associates the VRF with the Layer 3 interface.
Step 4	ipv6 enable Example: Switch(config-if)# ipv6 enable	Enable IPv6 on the interface.

	Command or Action	Purpose
Step 5	ipv6 address <i>ip-address subnet-mask</i> Example: Switch(config-if)# ipv6 address ip-address subnet-mask	Enters the IPv6 address for the interface.
Step 6	show ipv6 vrf [brief detail interfaces] [<i>vrf-name</i>] Example: Switch# show ipv6 vrf [brief detail interfaces] [vrf-name]	Verifies the configuration. Displays information about the configured VRFs.
Step 7	copy running-config startup-config Example: Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example

This example shows how to associate an interface to VRFs:

```
Switch(config-vrf)# interface ethernet0/1
Switch(config-if)# vrf forwarding red
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 address 5000::72B/64
```

Populate VRF with Routes via Routing Protocols

Configuring VRF Static Routes

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | **interface-type** *interface-number* [*ipv6-address*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	ipv6 route [vrf <i>vrf-name</i>] <i>ipv6-prefix/prefix-length</i> { <i>ipv6-address</i> interface-type <i>interface-number</i> [<i>ipv6-address</i>]} Example:	To configure static routes specific to VRF.

	Command or Action	Purpose
	Switch(config)# ipv6 route [vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address interface-type interface-number [ipv6-address]}	

Example

```
Switch(config)# ipv6 route vrf v6a 7000::/64 TenGigabitEthernet32 4000::2
```

Configuring OSPFv3 Router Process

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 process-id**
3. **area area-ID [default-cot | nssa | stub]**
4. **router-id router-id**
5. **address-family ipv6 unicast vrf vrf-name**
6. **redistribute source-protocol [process-id] options**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	router ospfv3 process-id Example: Switch(config)# router ospfv3 process-id	Enables OSPFv3 router configuration mode for the IPv6 address family.
Step 3	area area-ID [default-cot nssa stub] Example: Switch(config-router)# area area-ID [default-cot nssa stub]	Configures the OSPFv3 area.
Step 4	router-id router-id Example: Switch(config-router)# router-id router-id	Use a fixed router ID.
Step 5	address-family ipv6 unicast vrf vrf-name Example: Switch(config-router)# address-family ipv6 unicast vrf vrf-name	Enters IPv6 address family configuration mode for OSPFv3 in VRF vrf-name

	Command or Action	Purpose
Step 6	redistribute source-protocol [<i>process-id</i>] options Example: Switch(config-router)# redistribute source-protocol [<i>process-id</i>] options	Redistributes IPv6 routes from one routing domain into another routing domain.
Step 7	end Example: Switch(config-router)# end	Returns to privileged EXEC mode.

Example

This example shows how to configure the OSPFv3 router process:

```
Switch(config-router)# router ospfv3 1
Switch(config-router)# router-id 1.1.1.1
Switch(config-router)# address-family ipv6 unicast
Switch(config-router-af)# exit-address-family
```

Enabling OSPFv3 on an Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type-number*
3. **ospfv3** *process-id* **area** *area-ID* **ipv6** [**instance** *instance-id*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	interface <i>type-number</i> Example: Switch(config-vrf)# interface <i>type-number</i>	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 3	ospfv3 <i>process-id</i> area <i>area-ID</i> ipv6 [instance <i>instance-id</i>] Example: Switch(config-if)# ospfv3 <i>process-id</i> area <i>area-ID</i> <i>ipv6</i> [instance <i>instance-id</i>]	Enables OSPFv3 on an interface with IPv6 AF.
Step 4	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Switch(config-if)# end	

Example

This example show how to enable OSPFv3 on an interface:

```
Switch(config)# interface GigabitEthernet2/1
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 4000::2/64
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 ospf 1 area 0
Switch(config-if)# end
```

Configuring EIGRPv6 Routing Process

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *virtual-instance-name*
3. **address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*
4. **topology** {**base** | **topology-name** **tid** **number**
5. **exit-af****topology**
6. **eigrp router-id** *ip-address*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	router eigrp <i>virtual-instance-name</i> Example: Switch(config)# router eigrp virtual-instance-name	Configures the EIGRP routing process and enters router configuration mode.
Step 3	address-family ipv6 vrf <i>vrf-name</i> autonomous-system <i>autonomous-system-number</i> Example: Switch(config-router)# address-family ipv6 vrf vrf-name autonomous-system autonomous-system-number	Enables EIGRP IPv6 VRF-Lite and enters address family configuration mode.
Step 4	topology { base topology-name tid number Example: Switch(config-router-af)# topology {base topology-name tid number	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.

	Command or Action	Purpose
Step 5	exit-aftopology Example: Switch(config-router-af-topology)# exit-aftopology	Exits address family topology configuration mode.
Step 6	eigrp router-id <i>ip-address</i> Example: Switch(config-router)# eigrp router-id ip-address	Enables the use of a fixed router-id.
Step 7	end Example: Switch(config-router)# end	Exits router configuration mode.

Example

This example shows how to configure an EIGRP routing process:

```
Switch(config)# router eigrp test
Switch(config-router)# address-family ipv6 unicast vrf b1 autonomous-system 10
Switch(config-router-af)# topology base
Switch(config-router-af-topology)# exit-af-topology
Switch(config-router)# eigrp router-id 2.3.4.5
Switch(config-router)# exit-address-family
```

Configuring EIGRPv6 Routing Process**SUMMARY STEPS**

1. **configure terminal**
2. **router bgp *as-number***
3. **neighbor *peer-group-name peer-group***
4. **neighbor {*ip-address* | *ipv6-address*[%] | *peer-group-name*} remote-as *autonomous-system-number* [*alternate-as autonomous-system-number ...*]**
5. **address-family ipv6 [*vrf vrf-name*] [*unicast* | *multicast* | *vpn6*]**
6. **neighbor *ipv6-address peer-group peer-group-name***
7. **neighbor {*ip-address* | *peer-group-name* | *ipv6-address*[%]} route-map *map-name* {*in* | *out*}**
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 2	router bgp <i>as-number</i> Example: Switch(config)# router bgp as-number	Enters router configuration mode for the specified routing process.
Step 3	neighbor <i>peer-group-name peer-group</i> Example: Switch(config-router)# neighbor peer-group-name peer-group	Creates a multiprotocol BGP peer group.
Step 4	neighbor { <i>ip-address ipv6-address[%] peer-group-name</i> } remote-as <i>autonomous-system-number</i> [alternate-as <i>autonomous-system-number ...</i>] Example: Switch(config-router)# neighbor {ip-address ipv6-address[%] peer-group-name} remote-as autonomous-system-number [alternate-as autonomous-system-number ...]	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 5	address-family ipv6 [vrf <i>vrf-name</i>] [unicast multicast vpv6] Example: Switch(config-router)# address-family ipv6 [vrf vrf-name] [unicast multicast vpv6]	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the switch is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.
Step 6	neighbor <i>ipv6-address peer-group peer-group-name</i> Example: Switch(config-router-af)# neighbor ipv6-address peer-group peer-group-name	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 7	neighbor { <i>ip-address peer-group-name ipv6-address[%]</i> } route-map <i>map-name</i> { in out } Example: Switch(config-router-af)# neighbor {ip-address peer-group-name ipv6-address[%]} route-map map-name {in out}	Applies a route map to incoming or outgoing routes. Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.
Step 8	exit Example: Switch(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode.

Example

This example shows how to configure EBRIPv6:

```

Switch(config)# router bgp 2
Switch(config-router)# bgp router-id 2.2.2.2
Switch(config-router)# bgp log-neighbor-changes
Switch(config-router)# no bgp default ipv4-unicast
Switch(config-router)# neighbor 2500::1 remote-as 1
Switch(config-router)# neighbor 4000::2 remote-as 3
Switch(config-router)# address-family ipv6 vrf b1
Switch(config-router-af)# network 2500::/64
Switch(config-router-af)# network 4000::/64
Switch(config-router-af)# neighbor 2500::1 remote-as 1
Switch(config-router-af)# neighbor 2500::1 activate
Switch(config-router-af)# neighbor 4000::2 remote-as 3
Switch(config-router-af)# neighbor 4000::2 activate
Switch(config-router-af)# exit-address-family

```

Additional Information for VRF-lite

VPN Co-existence Between IPv4 and IPv6

Backward compatibility between the “older” CLI for configuring IPv4 and the “new” CLI for IPv6 exists. This means that a configuration might contain both CLI. The IPv4 CLI retains the ability to have on the same interface, an IP address defined within a VRF as well as an IPv6 address defined in the global routing table.

For example:

```

vrf definition red
  rd 100:1
  address family ipv6
  route-target both 200:1
  exit-address-family
!
ip vrf blue
  rd 200:1
  route-target both 200:1
!
interface Ethernet0/0
  vrf forwarding red
  ip address 50.1.1.2 255.255.255.0
  ipv6 address 4000::72B/64
!
interface Ethernet0/1
  vrf forwarding blue
  ip address 60.1.1.2 255.255.255.0
  ipv6 address 5000::72B/64

```

In this example, all addresses (v4 and v6) defined for Ethernet0/0 refer to VRF red whereas for Ethernet0/1, the IP address refers to VRF blue but the ipv6 address refers to the global IPv6 routing table.

Verifying VRF-lite Configuration

Displaying IPv4 VRF-lite Status

To display information about VRF-lite configuration and status, perform one of the following tasks:

Command	Purpose
Switch# show ip protocols vrf <i>vrf-name</i>	Displays routing protocol information associated with a VRF.
Switch# show ip route vrf <i>vrf-name</i> [connected] [<i>protocol</i>] [<i>as-number</i>] [list] [mobile] [odr] [profile] [static] [summary] [supernets-only]	Displays IP routing table information associated with a VRF.
Switch# show ip vrf [brief detail interfaces] [<i>vrf-name</i>]	Displays information about the defined VRF instances.
Switch# bidir vrf <i>instance-name a.b.c.d</i> active bidirectional count interface proxy pruned sparse ssm static summary	Displays information about the defined VRF instances.

This example shows how to display multicast route table information within a VRF instance:

```
Switch# show ip mroute 226.0.0.2
IP Multicast Routing Table
Flags: S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group, c - PFP-SA cache created entry
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 226.0.0.2), 00:01:17/stopped, RP 1.1.1.1, flags: SJCF
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36

(5.0.0.11, 226.0.0.2), 00:01:17/00:01:42, flags: FT
  Incoming interface: Vlan5, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan100, Forward/Sparse, 00:01:17/00:02:36
```

Displaying IPv6 VRF-lite Status

To display information about VRF-lite configuration and status, perform one of the following tasks:

Command	Purpose
	Displays routing protocol information associated with a VRF.
Switch# show ipv6 mfib vrf <i>instance-name</i> <i>a.b.c.d</i> active all count linkscope route summary update-sets verbose	Displays information about the defined VRF instances.

This example shows how to display multicast route table information within a VRF instance:

```

show ipv6 mroute vrf vrf1 FF05:ABCD:0:0:0:0:1
Multicast Routing Table
Flags: S - Sparse, B - Bidir Group, s - SSM Group,
C - Connected, L - Local, I - Received Source Specific Host Report,
P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
J - Join SPT, Y - Joined MDT-data group,
y - Sending to MDT-data group

g - BGP signal originated, G - BGP Signal received,
N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
q - BGP Src-Active originated, Q - BGP Src-Active received
E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State

(*, FF05:ABCD::1), 00:06:22/never, RP 1010:ABCD::10, flags: SCJ
Incoming interface: Port-channel33
RPF nbr: FE80::2E31:24FF:FE06:134A
Immediate outgoing interface list:
TenGigabitEthernet4/0/18, Forward, 00:06:22/never

(3232:ABCD::2, FF05:ABCD::1), 00:04:54/00:02:16, flags: SJT
Incoming interface: Port-channel33
RPF nbr: FE80::2E31:24FF:FE06:134A
Inherited outgoing interface list:
TenGigabitEthernet4/0/18, Forward, 00:06:22/never
    
```

The following example displays the output of **show ipv6 mfib** command.

```

Switch# show ipv6 mfib vrf vrf1 FF05:ABCD:0:0:0:0:1
Entry Flags:      C - Directly Connected, S - Signal, IA - Inherit A flag,
                  ET - Data Rate Exceeds Threshold, K - Keepalive
                  DDE - Data Driven Event, HW - Hardware Installed
                  ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                  MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                  MS - MoFRR Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags:  IC - Internal Copy, NP - Not platform switched,
                  NS - Negate Signalling, SP - Signal Present,
                  A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                  MA - MFIB Accept, A2 - Accept backup,
                  RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   FS Pkt Count/PS Pkt Count
    
```

```

VRF testvrf1
(*,FF05:ABCD::1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 295/0/512/0, Other: 0/0/0
Port-channel33 Flags: A NS
TenGigabitEthernet4/0/18 Flags: F NS
Pkts: 0/0
(3232:ABCD::2,FF05:ABCD::1) Flags: HW
SW Forwarding: 50/0/512/0, Other: 111/0/111
HW Forwarding: 4387686/14849/512/59398, Other: 0/0/0
Port-channel33 Flags: A
TenGigabitEthernet4/0/18 Flags: F NS
Pkts: 0/50

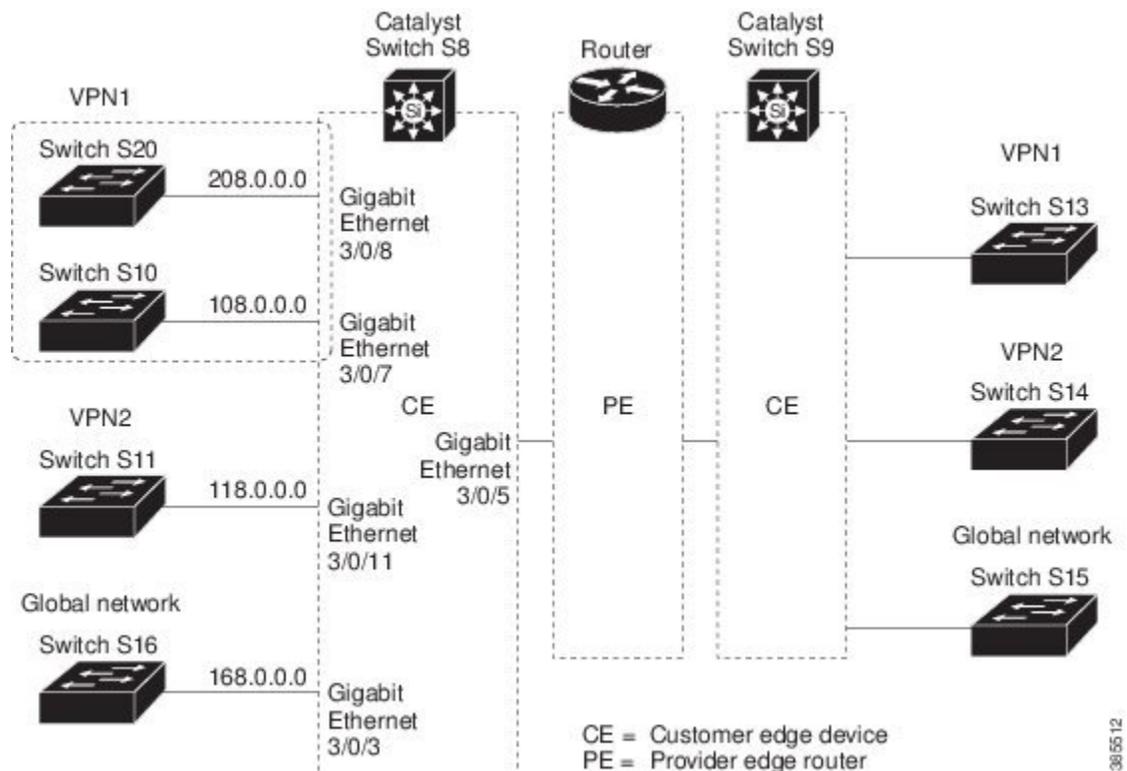
Switch#
    
```

Configuration Examples for VRF-lite

Configuration Example for IPv4 VRF-lite

OSPF is the protocol used in VPN1, VPN2, and the global network. BGP is used in the CE to PE connections. The example commands show how to configure the CE switch S8 and include the VRF configuration for switches S20 and S10 and the PE router commands related to traffic with switch S8. Commands for configuring the other switches are not included but would be similar.

Figure 5: VRF-lite Configuration Example



Configuring Switch S8

On switch S8, enable routing and configure VRF.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# ip vrf v11
Switch(config-vrf)# rd 800:1
Switch(config-vrf)# route-target export 800:1
Switch(config-vrf)# route-target import 800:1
Switch(config-vrf)# exit
Switch(config)# ip vrf v12
Switch(config-vrf)# rd 800:2
Switch(config-vrf)# route-target export 800:2
Switch(config-vrf)# route-target import 800:2
Switch(config-vrf)# exit
```

Configure the loopback and physical interfaces on switch S8. Fast Ethernet interface 3/5 is a trunk connection to the PE. Interfaces 3/7 and 3/11 connect to VPNs:

```
Switch(config)# interface loopback1
Switch(config-if)# vrf forwarding v11
Switch(config-if)# ip address 8.8.1.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface loopback2
Switch(config-if)# vrf forwarding v12
Switch(config-if)# ip address 8.8.2.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/5
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/8
Switch(config-if)# switchport access vlan 208
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet3/11
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit
```

Configure the VLANs used on switch S8. VLAN 10 is used by VRF 11 between the CE and the PE. VLAN 20 is used by VRF 12 between the CE and the PE. VLANs 118 and 208 are used for VRF for the VPNs that include switch S11 and switch S20, respectively:

```
Switch(config)# interface Vlan10
Switch(config-if)# vrf forwarding v11
Switch(config-if)# ip address 38.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan20
Switch(config-if)# vrf forwarding v12
Switch(config-if)# ip address 83.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# vrf forwarding v12
Switch(config-if)# ip address 118.0.0.8 255.255.255.0
```

```
Switch(config-if)# exit

Switch(config)# interface Vlan208
Switch(config-if)# vrf forwarding v11
Switch(config-if)# ip address 208.0.0.8 255.255.255.0
Switch(config-if)# exit
```

Configure OSPF routing in VPN1 and VPN2:

```
Switch(config)# router ospf 1 vrf v11
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
Switch(config)# router ospf 2 vrf v12
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
```

Configure BGP for CE to PE routing:

```
Switch(config)# router bgp 800
Switch(config-router)# address-family ipv4 vrf v12
Switch(config-router-af)# redistribute ospf 2 match internal
Switch(config-router-af)# neighbor 83.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 83.0.0.3 activate
Switch(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Switch(config-router-af)# exit
```

```
Switch(config-router)# address-family ipv4 vrf v11
Switch(config-router-af)# redistribute ospf 1 match internal
Switch(config-router-af)# neighbor 38.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 38.0.0.3 activate
Switch(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Switch(config-router-af)# end
```

Configuring Switch S20

Configure S20 to connect to CE:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Fast Ethernet 0/7
Switch(config-if)# no switchport
Switch(config-if)# ip address 208.0.0.20 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring Switch S11

Configure S11 to connect to CE:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Gigabit Ethernet 0/3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface Vlan118
```

```
Switch(config-if)# ip address 118.0.0.11 255.255.255.0
Switch(config-if)# exit
```

```
Switch(config)# router ospf 101
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring the PE Switch S3

On switch S3 (the router), these commands configure only the connections to switch S8:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit

Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Loopback2
Router(config-if)# vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

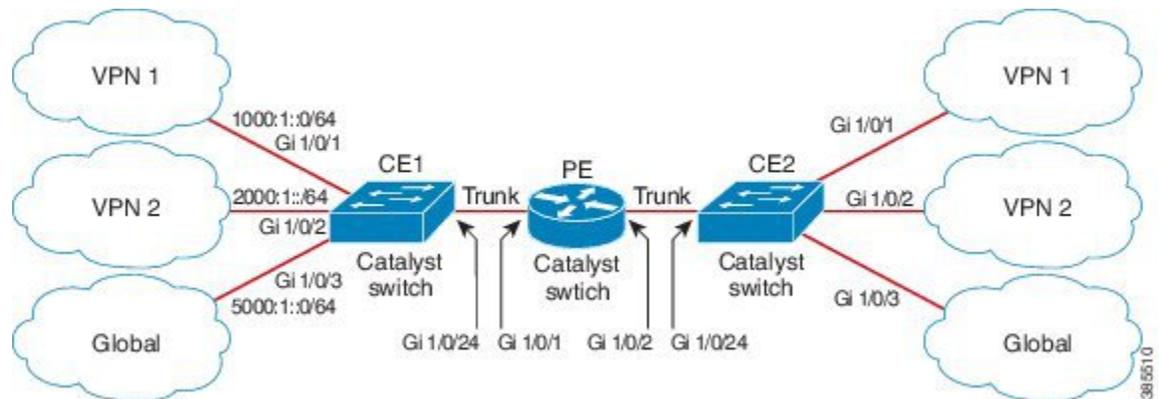
Router(config)# interface Fast Ethernet3/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf v1
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

Configuration Example for IPv6 VRF-lite

The following topology illustrates how to use OSPFv3 for CE-PE routing.

Figure 6: VRF-lite Configuration Example



Configuring CE1 Switch

```

ipv6 unicast-routing
vrf definition v1
 rd 100:1
 !
address-family ipv6
 exit-address-family
 !

vrf definition v2
 rd 200:1
 !
address-family ipv6
 exit-address-family
 !

interface Vlan100
 vrf forwarding v1
 ipv6 address 1000:1::1/64
 ospfv3 100 ipv6 area 0
 !

interface Vlan200
 vrf forwarding v2
 ipv6 address 2000:1::1/64
 ospfv3 200 ipv6 area 0
 !

interface GigabitEthernet 1/0/1
 switchport access vlan 100
 end

interface GigabitEthernet 1/0/2
 switchport access vlan 200
 end

interface GigabitEthernet 1/0/24
 switchport trunk encapsulation dot1q

```

```

switchport mode trunk
end

router ospfv3 100
router-id 10.10.10.10
!
address-family ipv6 unicast vrf v1
redistribute connected
area 0 normal
exit-address-family
!

router ospfv3 200
router-id 20.20.20.20
!
address-family ipv6 unicast vrf v2
redistribute connected
area 0 normal
exit-address-family
!

```

Configuring PE Switch

```

ipv6 unicast-routing

vrf definition v1
rd 100:1
!
address-family ipv6
exit-address-family
!

vrf definition v2
rd 200:1
!
address-family ipv6
exit-address-family
!

interface Vlan600
vrf forwarding v1
no ipv6 address
ipv6 address 1000:1::2/64
ospfv3 100 ipv6 area 0
!

interface Vlan700
vrf forwarding v2
no ipv6 address
ipv6 address 2000:1::2/64
ospfv3 200 ipv6 area 0
!

interface Vlan800
vrf forwarding v1
ipv6 address 3000:1::7/64
ospfv3 100 ipv6 area 0
!

interface Vlan900
vrf forwarding v2
ipv6 address 4000:1::7/64
ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1

```

```

switchport trunk encapsulation dot1q
switchport mode trunk
exit

interface GigabitEthernet 1/0/2
switchport trunk encapsulation dot1q

switchport mode trunk
exit

router ospfv3 100
router-id 30.30.30.30
!
address-family ipv6 unicast vrf v1
redistribute connected
area 0 normal
exit-address-family
!
address-family ipv6 unicast vrf v2
redistribute connected
area 0 normal
exit-address-family
!

```

Configuring CE2 Switch

```

ipv6 unicast-routing

vrf definition v1
rd 100:1
!
address-family ipv6
exit-address-family
!

vrf definition v2
rd 200:1
!
address-family ipv6
exit-address-family
!

interface Vlan100
vrf forwarding v1

ipv6 address 1000:1::3/64
ospfv3 100 ipv6 area 0
!

interface Vlan200
vrf forwarding v2
ipv6 address 2000:1::3/64
ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
switchport access vlan 100
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24

```

```
switchport trunk encapsulation dot1q
switchport mode trunk
end

router ospfv3 100
router-id 40.40.40.40
!
address-family ipv6 unicast vrf v1
redistribute connected
area 0 normal
exit-address-family
!

router ospfv3 200
router-id 50.50.50.50
!
address-family ipv6 unicast vrf v2
redistribute connected

area 0 normal
exit-address-family
!
```



CHAPTER 6

Configuring BGP

- [Information About BGP, on page 101](#)
- [How to Configure BGP, on page 113](#)
- [Configuration Examples for BGP, on page 150](#)
- [Monitoring and Maintaining BGP, on page 151](#)

Information About BGP

The Border Gateway Protocol (BGP) is an exterior gateway protocol used to set up an interdomain routing system that guarantees the loop-free exchange of routing information between autonomous systems. Autonomous systems are made up of routers that operate under the same administration and that run Interior Gateway Protocols (IGPs), such as RIP or OSPF, within their boundaries and that interconnect by using an Exterior Gateway Protocol (EGP). BGP Version 4 is the standard EGP for interdomain routing in the Internet. The protocol is defined in RFCs 1163, 1267, and 1771.

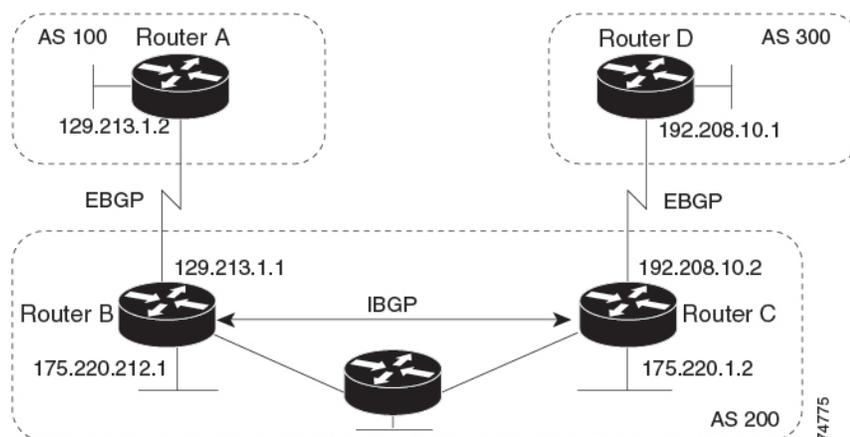


Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

BGP Network Topology

Routers that belong to the same autonomous system (AS) and that exchange BGP updates run internal BGP (IBGP), and routers that belong to different autonomous systems and that exchange BGP updates run external BGP (EBGP). Most configuration commands are the same for configuring EBGP and IBGP. The difference is that the routing updates are exchanged either between autonomous systems (EBGP) or within an AS (IBGP). The figure given below shows a network that is running both EBGP and IBGP.

Figure 7: EBGP, IBGP, and Multiple Autonomous Systems



Before exchanging information with an external AS, BGP ensures that networks within the AS can be reached by defining internal BGP peering among routers within the AS and by redistributing BGP routing information to IGPs that run within the AS, such as IGRP and OSPF.

Routers that run a BGP routing process are often referred to as BGP speakers. BGP uses the Transmission Control Protocol (TCP) as its transport protocol (specifically port 179). Two BGP speakers that have a TCP connection to each other for exchanging routing information are known as peers or neighbors. In the above figure, Routers A and B are BGP peers, as are Routers B and C and Routers C and D. The routing information is a series of AS numbers that describe the full path to the destination network. BGP uses this information to construct a loop-free map of autonomous systems.

The network has these characteristics:

- Routers A and B are running EBGP, and Routers B and C are running IBGP. Note that the EBGP peers are directly connected and that the IBGP peers are not. As long as there is an IGP running that allows the two neighbors to reach one another, IBGP peers do not have to be directly connected.
- All BGP speakers within an AS must establish a peer relationship with each other. That is, the BGP speakers within an AS must be fully meshed logically. BGP4 provides two techniques that reduce the requirement for a logical full mesh: confederations and route reflectors.
- AS 200 is a transit AS for AS 100 and AS 300—that is, AS 200 is used to transfer packets between AS 100 and AS 300.

BGP peers initially exchange their full BGP routing tables and then send only incremental updates. BGP peers also exchange keepalive messages (to ensure that the connection is up) and notification messages (in response to errors or special conditions).

In BGP, each route consists of a network number, a list of autonomous systems that information has passed through (the autonomous system path), and a list of other path attributes. The primary function of a BGP system is to exchange network reachability information, including information about the list of AS paths, with other BGP systems. This information can be used to determine AS connectivity, to prune routing loops, and to enforce AS-level policy decisions.

A router or device running Cisco IOS does not select or use an IBGP route unless it has a route available to the next-hop router and it has received synchronization from an IGP (unless IGP synchronization is disabled). When multiple routes are available, BGP bases its path selection on attribute values. See the “Configuring BGP Decision Attributes” section for information about BGP attributes.

BGP Version 4 supports classless interdomain routing (CIDR) so you can reduce the size of your routing tables by creating aggregate routes, resulting in supernets. CIDR eliminates the concept of network classes within BGP and supports the advertising of IP prefixes.

Nonstop Forwarding Awareness

To enable this feature with BGP routing, you need to enable Graceful Restart. When the neighboring router is NSF-capable, and this feature is enabled, the Layer 3 device continues to forward packets from the neighboring router, during the interval when the primary Route Processor (RP) in a device is failing and the backup RP is taking over, or while the primary RP is manually reloaded for a nondisruptive software upgrade.

Information About BGP Routing

To enable BGP routing, you establish a BGP routing process and define the local network. Because BGP must completely recognize the relationships with its neighbors, you must also specify a BGP neighbor.

BGP supports two kinds of neighbors: internal and external. Internal neighbors are in the same AS; external neighbors are in different autonomous systems. External neighbors are usually adjacent to each other and share a subnet, but internal neighbors can be anywhere in the same AS.

The switch supports the use of private AS numbers, usually assigned by service providers and given to systems whose routes are not advertised to external neighbors. The private AS numbers are from 64512 to 65535. You can configure external neighbors to remove private AS numbers from the AS path by using the **neighbor remove-private-as** router configuration command. Then when an update is passed to an external neighbor, if the AS path includes private AS numbers, these numbers are dropped.

If your AS will be passing traffic through it from another AS to a third AS, it is important to be consistent about the routes it advertises. If BGP advertised a route before all routers in the network had learned about the route through the IGP, the AS might receive traffic that some routers could not yet route. To prevent this from happening, BGP must wait until the IGP has propagated information across the AS so that BGP is synchronized with the IGP. Synchronization is enabled by default. If your AS does not pass traffic from one AS to another AS, or if all routers in your autonomous systems are running BGP, you can disable synchronization, which allows your network to carry fewer routes in the IGP and allows BGP to converge more quickly.

Routing Policy Changes

Routing policies for a peer include all the configurations that might affect inbound or outbound routing table updates. When you have defined two routers as BGP neighbors, they form a BGP connection and exchange routing information. If you later change a BGP filter, weight, distance, version, or timer, or make a similar configuration change, you must reset the BGP sessions so that the configuration changes take effect.

There are two types of reset, hard reset and soft reset. Cisco IOS Releases 12.1 and later support a soft reset without any prior configuration. To use a soft reset without preconfiguration, both BGP peers must support the soft route refresh capability, which is advertised in the OPEN message sent when the peers establish a TCP session. A soft reset allows the dynamic exchange of route refresh requests and routing information between BGP routers and the subsequent re-advertisement of the respective outbound routing table.

- When soft reset generates inbound updates from a neighbor, it is called dynamic inbound soft reset.
- When soft reset sends a set of updates to a neighbor, it is called outbound soft reset.

A soft inbound reset causes the new inbound policy to take effect. A soft outbound reset causes the new local outbound policy to take effect without resetting the BGP session. As a new set of updates is sent during outbound policy reset, a new inbound policy can also take effect.

The table that is given below lists the advantages and disadvantages hard reset and soft reset.

Table 6: Advantages and Disadvantages of Hard and Soft Resets

Type of Reset	Advantages	Disadvantages
Hard reset	No memory overhead	The prefixes in the BGP, IP, and FIB tables provided by the neighbor are lost. Not recommended.
Outbound soft reset	No configuration, no storing of routing table updates	Does not reset inbound routing table updates
Dynamic inbound soft reset	Does not clear the BGP session and cache Does not require storing of routing table updates and has no memory overhead	Both BGP routers must support the soft reset capability (in Cisco IOS Release 12.1 and later)

BGP Decision Attributes

When a BGP speaker receives updates from multiple autonomous systems that describe different paths to the same destination, it must choose the single best path for reaching that destination. When chosen, the selected path is entered into the BGP routing table and propagated to its neighbors. The decision is based on the value of attributes that the update contains and other BGP-configurable factors.

When a BGP peer learns two EBGP paths for a prefix from a neighboring AS, it chooses the best path and inserts that path in the IP routing table. If BGP multipath support is enabled and the EBGP paths are learned from the same neighboring autonomous systems, instead of a single best path, multiple paths are installed in the IP routing table. Then, during packet switching, per-packet or per-destination load-balancing is performed among the multiple paths. The **maximum-paths** router configuration command controls the number of paths allowed.

These factors summarize the order in which BGP evaluates the attributes for choosing the best path:

1. If the path specifies a next hop that is inaccessible, drop the update. The BGP next-hop attribute, automatically determined by the software, is the IP address of the next hop that is going to be used to reach a destination. For EBGP, this is usually the IP address of the neighbor that is specified by the **neighbor remote-as router** configuration command. You can disable next-hop processing by using route maps or the **neighbor next-hop-self** router configuration command.
2. Prefer the path with the largest weight (a Cisco proprietary parameter). The weight attribute is local to the router and not propagated in routing updates. By default, the weight attribute is 32768 for paths that the router originates and zero for other paths. Routes with the largest weight are preferred. You can use access lists, route maps, or the **neighbor weight** router configuration command to set weights.
3. Prefer the route with the highest local preference. Local preference is part of the routing update and exchanged among routers in the same AS. The default value of the local preference attribute is 100. You can set local preference by using the **bgp default local-preference** router configuration command or by using a route map.
4. Prefer the route that was originated by BGP running on the local router.

5. Prefer the route with the shortest AS path.
6. Prefer the route with the lowest origin type. An interior route or IGP is lower than a route learned by EGP, and an EGP-learned route is lower than one of unknown origin or learned in another way.
7. Prefer the route with the lowest multi-exit discriminator (MED) metric attribute if the neighboring AS is the same for all routes considered. You can configure the MED by using route maps or by using the **default-metric** router configuration command. When an update is sent to an IBGP peer, the MED is included.
8. Prefer the external (EBGP) path over the internal (IBGP) path.
9. Prefer the route that can be reached through the closest IGP neighbor (the lowest IGP metric). This means that the router will prefer the shortest internal path within the AS to reach the destination (the shortest path to the BGP next-hop).
10. If the following conditions are all true, insert the route for this path into the IP routing table:
 - Both the best route and this route are external.
 - Both the best route and this route are from the same neighboring autonomous system.
 - Maximum-paths is enabled.
11. If multipath is not enabled, prefer the route with the lowest IP address value for the BGP router ID. The router ID is usually the highest IP address on the router or the loopback (virtual) address, but might be implementation-specific.

Route Maps

Within BGP, route maps can be used to control and to modify routing information and to define the conditions by which routes are redistributed between routing domains. Each route map has a name that identifies the route map (*map tag*) and an optional sequence number.

BGP Filtering

You can filter BGP advertisements by using AS-path filters, such as the **as-path access-list** global configuration command and the **neighbor filter-list** router configuration command. You can also use access lists with the **neighbor distribute-list** router configuration command. Distribute-list filters are applied to network numbers. See the “Controlling Advertising and Processing in Routing Updates” section for information about the **distribute-list** command.

You can use route maps on a per-neighbor basis to filter updates and to modify various attributes. A route map can be applied to either inbound or outbound updates. Only the routes that pass the route map are sent or accepted in updates. On both inbound and outbound updates, matching is supported based on AS path, community, and network numbers. Autonomous system path matching requires the **match as-path access-list** route-map command, community based matching requires the **match community-list** route-map command, and network-based matching requires the **ip access-list** global configuration command.

Prefix List for BGP Filtering

You can use prefix lists as an alternative to access lists in many BGP route filtering commands, including the **neighbor distribute-list** router configuration command. The advantages of using prefix lists include performance improvements in loading and lookup of large lists, incremental update support, easier CLI configuration, and greater flexibility.

Filtering by a prefix list involves matching the prefixes of routes with those listed in the prefix list, as when matching access lists. When there is a match, the route is used. Whether a prefix is permitted or denied is based upon these rules:

- An empty prefix list permits all prefixes.
- An implicit deny is assumed if a given prefix does not match any entries in a prefix list.
- When multiple entries of a prefix list match a given prefix, the sequence number of a prefix list entry identifies the entry with the lowest sequence number.

By default, sequence numbers are generated automatically and incremented in units of five. If you disable the automatic generation of sequence numbers, you must specify the sequence number for each entry. You can specify sequence values in any increment. If you specify increments of one, you cannot insert additional entries into the list; if you choose large increments, you might run out of values.

BGP Community Filtering

One way that BGP controls the distribution of routing information based on the value of the COMMUNITIES attribute. The attribute is a way to group destinations into communities and to apply routing decisions based on the communities. This method simplifies configuration of a BGP speaker to control distribution of routing information.

A community is a group of destinations that share some common attribute. Each destination can belong to multiple communities. AS administrators can define to which communities a destination belongs. By default, all destinations belong to the general Internet community. The community is identified by the COMMUNITIES attribute, an optional, transitive, global attribute in the numerical range from 1 to 4294967200. These are some predefined, well-known communities:

- **internet**—Advertise this route to the Internet community. All routers belong to it.
- **no-export**—Do not advertise this route to EBGp peers.
- **no-advertise**—Do not advertise this route to any peer (internal or external).
- **local-as**—Do not advertise this route to peers outside the local autonomous system.

Based on the community, you can control which routing information to accept, prefer, or distribute to other neighbors. A BGP speaker can set, append, or modify the community of a route when learning, advertising, or redistributing routes. When routes are aggregated, the resulting aggregate has a COMMUNITIES attribute that contains all communities from all the initial routes.

You can use community lists to create groups of communities to use in a match clause of a route map. As with an access list, a series of community lists can be created. Statements are checked until a match is found. As soon as one statement is satisfied, the test is concluded.

BGP Neighbors and Peer Groups

Often many BGP neighbors are configured with the same update policies (that is, the same outbound route maps, distribute lists, filter lists, update source, and so on). Neighbors with the same update policies can be grouped into peer groups to simplify configuration and to make updating more efficient. When you have configured many peers, we recommend this approach.

To configure a BGP peer group, you create the peer group, assign options to the peer group, and add neighbors as peer group members. You configure the peer group by using the **neighbor** router configuration commands. By default, peer group members inherit all the configuration options of the peer group, including the remote-as (if configured), version, update-source, out-route-map, out-filter-list, out-dist-list, minimum-advertisement-interval, and next-hop-self. All peer group members also inherit changes that are made to the peer group. Members can also be configured to override the options that do not affect outbound updates.

Aggregate Routes

Classless interdomain routing (CIDR) enables you to create aggregate routes (or supernets) to minimize the size of routing tables. You can configure aggregate routes in BGP either by redistributing an aggregate route into BGP or by creating an aggregate entry in the BGP routing table. An aggregate address is added to the BGP table when there is at least one more specific entry in the BGP table.

Routing Domain Confederations

One way to reduce the IBGP mesh is to divide an autonomous system into multiple subautonomous systems and to group them into a single confederation that appears as a single autonomous system. Each autonomous system is fully meshed within itself and has a few connections to other autonomous systems in the same confederation. Even though the peers in different autonomous systems have EBGP sessions, they exchange routing information as if they were IBGP peers. Specifically, the next hop, MED, and local preference information are preserved. You can then use a single IGP for all of the autonomous systems.

BGP Route Reflectors

BGP requires that all of the IBGP speakers be fully meshed. When a router receives a route from an external neighbor, it must advertise it to all internal neighbors. To prevent a routing information loop, all IBGP speakers must be connected. The internal neighbors do not send routes that are learned from internal neighbors to other internal neighbors.

With route reflectors, all IBGP speakers need not be fully meshed because another method is used to pass learned routes to neighbors. When you configure an internal BGP peer to be a route reflector, it is responsible for passing IBGP learned routes to a set of IBGP neighbors. The internal peers of the route reflector are divided into two groups: client peers and nonclient peers (all the other routers in the autonomous system). A route reflector reflects routes between these two groups. The route reflector and its client peers form a cluster. The nonclient peers must be fully meshed with each other, but the client peers need not be fully meshed. The clients in the cluster do not communicate with IBGP speakers outside their cluster.

When the route reflector receives an advertised route, it takes one of these actions, depending on the neighbor:

- A route from an external BGP speaker is advertised to all clients and nonclient peers.
- A route from a nonclient peer is advertised to all clients.

- A route from a client is advertised to all clients and nonclient peers. Hence, the clients need not be fully meshed.

Usually a cluster of clients has a single route reflector, and the cluster is identified by the route reflector router ID. To increase redundancy and to avoid a single point of failure, a cluster might have more than one route reflector. In this case, all route reflectors in the cluster must be configured with the same 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster. All the route reflectors serving a cluster should be fully meshed and should have identical sets of client and nonclient peers.

Route Dampening

Route flap dampening is a BGP feature designed to minimize the propagation of flapping routes across an internetwork. A route is considered to be flapping when it is repeatedly available, then unavailable, then available, then unavailable, and so on. When route dampening is enabled, a numeric penalty value is assigned to a route when it flaps. When a route's accumulated penalties reach a configurable limit, BGP suppresses advertisements of the route, even if the route is running. The reuse limit is a configurable value that is compared with the penalty. If the penalty is less than the reuse limit, a suppressed route that is up is advertised again.

Dampening is not applied to routes that are learned by IBGP. This policy prevents the IBGP peers from having a higher penalty for routes external to the AS.

Conditional BGP Route Injection

Routes that are advertised through the BGP are commonly aggregated to minimize the number of routes that are used and reduce the size of global routing tables. However, common route aggregation can obscure more specific routing information that is more accurate but not necessary to forward packets to their destinations. Routing accuracy is obscured by common route aggregation because a prefix that represents multiple addresses or hosts over a large topological area cannot be accurately reflected in a single route. Cisco software provides several methods by which you can originate a prefix into BGP. Prior to the BGP conditional route injection feature, the existing methods included redistribution and using the **network** or **aggregate-address** command. However, these methods assume the existence of more specific routing information (matching the route to be originated) in either the routing table or the BGP table.

BGP conditional route injection allows you to originate a prefix into a BGP routing table without the corresponding match. This feature allows more specific routes to be generated based on administrative policy or traffic engineering information in order to provide more specific control over the forwarding of packets to these more specific routes, which are injected into the BGP routing table only if the configured conditions are met. Enabling this feature will allow you to improve the accuracy of common route aggregation by conditionally injecting or replacing less specific prefixes with more specific prefixes. Only prefixes that are equal to or more specific than the original prefix may be injected. BGP conditional route injection is enabled with the **bgp inject-map exist-map** command and uses two route maps (inject map and exist map) to install one (or more) more specific prefixes into a BGP routing table. The exist map specifies the prefixes that the BGP speaker will track. The inject map defines the prefixes that will be created and installed into the local BGP table.



Note Inject maps and exist maps will only match a single prefix per route map clause. To inject additional prefixes, you must configure additional route map clauses. If multiple prefixes are used, the first prefix that is matched will be used.

BGP Peer Templates

To address some of the limitations of peer groups such as configuration management, BGP peer templates were introduced to support the BGP update group configuration.

A peer template is a configuration pattern that can be applied to neighbors that share policies. Peer templates are reusable and support inheritance, which allows the network operator to group and apply distinct neighbor configurations for BGP neighbors that share policies. Peer templates also allow the network operator to define complex configuration patterns through the capability of a peer template to inherit a configuration from another peer template.

There are two types of peer templates:

- Peer session templates are used to group and apply the configuration of general session commands that are common to all address family and NLRI configuration modes.
- Peer policy templates are used to group and apply the configuration of commands that are applied within specific address families and NLRI configuration modes.

Peer templates improve the flexibility and enhance the capability of neighbor configuration. Peer templates also provide an alternative to peer group configuration and overcome some limitations of peer groups. BGP peer devices using peer templates also benefit from automatic update group configuration. With the configuration of the BGP peer templates and the support of the BGP dynamic update peer groups, the network operator no longer must configure peer groups in BGP and the network can benefit from improved configuration flexibility and faster convergence.



Note A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies from peer templates.

The following restrictions apply to the peer policy templates:

- A peer policy template can directly or indirectly inherit up to eight peer policy templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

Inheritance in Peer Templates

The inheritance capability is a key component of peer template operation. Inheritance in a peer template is similar to node and tree structures that are commonly found in general computing, for example, file and directory trees. A peer template can directly or indirectly inherit the configuration from another peer template. The directly inherited peer template represents the tree in the structure. The indirectly inherited peer template represents a node in the tree. Because each node also supports inheritance, branches can be created that apply the configurations of all indirectly inherited peer templates within a chain back to the directly inherited peer template or the source of the tree.

This structure eliminates the need to repeat configuration statements that are commonly reapplied to groups of neighbors because common configuration statements can be applied once and then indirectly inherited by peer templates that are applied to neighbor groups with common configurations. Configuration statements that are duplicated separately within a node and a tree are filtered out at the source of the tree by the directly

inherited template. A directly inherited template overwrites any indirectly inherited statements that are duplicated in the directly inherited template.

Inheritance expands the scalability and flexibility of neighbor configuration by allowing you to chain together peer templates configurations to create simple configurations that inherit common configuration statements or complex configurations that apply specific configuration statements along with common inherited configurations. Specific details about configuring inheritance in peer session templates and peer policy templates are provided in the following sections.

When BGP neighbors use inherited peer templates, it can be difficult to determine which policies are associated with a specific template. The **detail** keyword of the **show ip bgp template peer-policy** command displays the detailed configuration of local and inherited policies that are associated with a specific template.

Peer Session Templates

Peer session templates are used to group and apply the configuration of general session commands to groups of neighbors that share session configuration elements. General session commands that are common for neighbors that are configured in different address families can be configured within the same peer session template. Peer session templates are created and configured in peer session configuration mode. Only general session commands can be configured in a peer session template. The following general session commands are supported by peer session templates:

- **description**
- **disable-connected-check**
- **ebgp-multihop**
- **exit peer-session**
- **inherit peer-session**
- **local-as**
- **password**
- **remote-as**
- **shutdown**
- **timers**
- **translate-update**
- **update-source**
- **version**

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template or through indirect inheritance from a peer session template. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an autonomous system.

Peer session templates support direct and indirect inheritance. A peer can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template.



Note If you attempt to configure more than one inherit statement with a single peer session template, an error message will be displayed.

This behavior allows a BGP neighbor to directly inherit only one session template and indirectly inherit up to seven additional peer session templates. This allows you to apply up to a maximum of eight peer session configurations to a neighbor: the configuration from the directly inherited peer session template and the configurations from up to seven indirectly inherited peer session templates. Inherited peer session configurations are evaluated first and applied starting with the last node in the branch and ending with the directly applied peer session template configuration at the source of the tree. The directly applied peer session template will have priority over inherited peer session template configurations. Any configuration statements that are duplicated in inherited peer session templates will be overwritten by the directly applied peer session template. So, if a general session command is reapplied with a different value, the subsequent value will have priority and overwrite the previous value that was configured in the indirectly inherited template. The following examples illustrate the use of this feature.

In the following example, the general session command **remote-as 1** is applied in the peer session template named SESSION-TEMPLATE-ONE:

```
template peer-session SESSION-TEMPLATE-ONE
  remote-as 1
  exit peer-session
```

Peer session templates support only general session commands. BGP policy configuration commands that are configured only for a specific address family or NLRI configuration mode are configured with peer policy templates.

Peer Policy Templates

Peer policy templates are used to group and apply the configuration of commands that are applied within specific address families and NLRI configuration mode. Peer policy templates are created and configured in peer policy configuration mode. BGP policy commands that are configured for specific address families are configured in a peer policy template. The following BGP policy commands are supported by peer policy templates:

- **advertisement-interval**
- **allowas-in**
- **as-override**
- **capability**
- **default-originate**
- **distribute-list**
- **dmzlink-bw**
- **exit-peer-policy**
- **filter-list**
- **inherit peer-policy**

- **maximum-prefix**
- **next-hop-self**
- **next-hop-unchanged**
- **prefix-list**
- **remove-private-as**
- **route-map**
- **route-reflector-client**
- **send-community**
- **send-label**
- **soft-reconfiguration**
- **unsuppress-map**
- **weight**

Peer policy templates are used to configure BGP policy commands that are configured for neighbors that belong to specific address families. Like peer session templates, peer policy templates are configured once and then applied to many neighbors through the direct application of a peer policy template or through inheritance from peer policy templates. The configuration of peer policy templates simplifies the configuration of BGP policy commands that are applied to all neighbors within an autonomous system.

Like a peer session template, a peer policy template supports inheritance. However, there are minor differences. A directly applied peer policy template can directly or indirectly inherit configurations from up to seven peer policy templates. So, a total of eight peer policy templates can be applied to a neighbor or neighbor group. Like route maps, inherited peer policy templates are configured with sequence numbers. Also like a route map, an inherited peer policy template is evaluated starting with the **inherit peer-policy** statement with the lowest sequence number and ending with the highest sequence number. However, there is a difference; a peer policy template will not collapse like a route map. Every sequence is evaluated, and if a BGP policy command is reapplied with a different value, it will overwrite any previous value from a lower sequence number.

The directly applied peer policy template and the **inherit peer-policy** statement with the highest sequence number will always have priority and be applied last. Commands that are reapplied in subsequent peer templates will always overwrite the previous values. This behavior is designed to allow you to apply common policy configurations to large neighbor groups and specific policy configurations only to certain neighbors and neighbor groups without duplicating individual policy configuration commands.

Peer policy templates support only policy configuration commands. BGP policy configuration commands that are configured only for specific address families are configured with peer policy templates.

The configuration of peer policy templates simplifies and improves the flexibility of BGP configuration. A specific policy can be configured once and referenced many times. Because a peer policy supports up to eight levels of inheritance, very specific and very complex BGP policies can also be created.

How to Configure BGP

Default BGP Configuration

The table given below shows the basic default BGP configuration.

Table 7. Default BGP Configuration

Feature	Default Setting
Aggregate address	Disabled: None defined.
AS path access list	None defined.
Auto summary	Disabled.
Best path	<ul style="list-style-type: none"> The router considers <i>as-path</i> in choosing a route and does not compare s from external BGP peers. Compare router ID: Disabled.
BGP community list	<ul style="list-style-type: none"> Number: None defined. When you permit a value for the community number defaults to an implicit deny for everything else that has not been permitted. Format: Cisco default format (32-bit number).
BGP confederation identifier/peers	<ul style="list-style-type: none"> Identifier: None configured. Peers: None identified.
BGP Fast external fallover	Enabled.
BGP local preference	100. The range is 0 to 4294967295 with the higher value preferred.
BGP network	None specified; no backdoor route advertised.
BGP route dampening	Disabled by default. When enabled: <ul style="list-style-type: none"> Half-life is 15 minutes. Re-use is 750 (10-second increments). Suppress is 2000 (10-second increments). Max-suppress-time is 4 times half-life; 60 minutes.
BGP router ID	The IP address of a loopback interface if one is configured or the highest IP address for a physical interface on the router.
Default information originate (protocol or network redistribution)	Disabled.

Feature	Default Setting
Default metric	Built-in, automatic metric translations.
Distance	<ul style="list-style-type: none"> • External route administrative distance: 20 (acceptable values are from 1 to 255) • Internal route administrative distance: 200 (acceptable values are from 1 to 255) • Local route administrative distance: 200 (acceptable values are from 1 to 255)
Distribute list	<ul style="list-style-type: none"> • In (filter networks received in updates): Disabled. • Out (suppress networks from being advertised in updates): Disabled.
Internal route redistribution	Disabled.
IP prefix list	None defined.
Multi exit discriminator (MED)	<ul style="list-style-type: none"> • Always compare: Disabled. Does not compare MEDs for paths from neighbors in different autonomous systems. • Best path compare: Disabled. • MED missing as worst path: Disabled. • Deterministic MED comparison is disabled.

Feature	Default Setting
Neighbor	<ul style="list-style-type: none"> • Advertisement interval: 30 seconds for external peers; 5 seconds for internal peers. • Change logging: Enabled. • Conditional advertisement: Disabled. • Default originate: No default route is sent to the neighbor. • Description: None. • Distribute list: None defined. • External BGP multihop: Only directly connected neighbors are allowed. • Filter list: None used. • Maximum number of prefixes received: No limit. • Next hop (router as next hop for BGP neighbor): Disabled. • Password: Disabled. • Peer group: None defined; no members assigned. • Prefix list: None specified. • Remote AS (add entry to neighbor BGP table): No peers defined. • Private AS number removal: Disabled. • Route maps: None applied to a peer. • Send community attributes: None sent to neighbors. • Shutdown or soft reconfiguration: Not enabled. • Timers: keepalive: 60 seconds; holdtime: 180 seconds. • Update source: Best local address. • Version: BGP Version 4. • Weight: Routes learned through BGP peer: 0; routes sourced by the local router: 32768.
NSF ¹ Awareness	Disabled ² . If enabled, allows Layer 3 switches to continue forwarding packets to neighboring NSF-capable router during hardware or software changes.
Route reflector	None configured.
Synchronization (BGP and IGP)	Disabled.
Table map update	Disabled.
Timers	Keepalive: 60 seconds; holdtime: 180 seconds.

¹ Nonstop Forwarding

² NSF Awareness can be enabled for IPv4 on switches with the license by enabling Graceful Restart.

Enabling BGP Routing

Before you begin



Note To enable BGP, the switch or stack master must be running the

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip routing Example: Device(config)# ip routing	Enables IP routing.
Step 4	router bgp <i>autonomous-system</i> Example: Device(config)# router bgp 45000	Enables a BGP routing process, assign it an AS number, and enter router configuration mode. The AS number can be from 1 to 65535, with 64512 to 65535 designated as private autonomous numbers.
Step 5	network <i>network-number</i> [mask <i>network-mask</i>] [route-map <i>route-map-name</i>] Example: Device(config)# network 10.108.0.0	Configures a network as local to this AS, and enter it in the BGP table.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>number</i> Example: Device(config)# neighbor 10.108.1.2 remote-as 65200	Adds an entry to the BGP neighbor table specifying that the neighbor identified by the IP address belongs to the specified AS. For EBGP, neighbors are usually directly connected, and the IP address is the address of the interface at the other end of the connection. For IBGP, the IP address can be the address of any of the router interfaces.

	Command or Action	Purpose
Step 7	neighbor {ip-address peer-group-name} remove-private-as Example: <pre>Device(config)# neighbor 172.16.2.33 remove-private-as</pre>	(Optional) Removes private AS numbers from the AS-path in outbound routing updates.
Step 8	synchronization Example: <pre>Device(config)# synchronization</pre>	(Optional) Enables synchronization between BGP and an IGP.
Step 9	auto-summary Example: <pre>Device(config)# auto-summary</pre>	(Optional) Enables automatic network summarization. When a subnet is redistributed from an IGP into BGP, only the network route is inserted into the BGP table.
Step 10	bgp graceful-restart Example: <pre>Device(config)# bgp graceful-start</pre>	(Optional) Enables NSF awareness on switch. By default, NSF awareness is disabled.
Step 11	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 12	show ip bgp network network-number Example: <pre>Device# show ip bgp network 10.108.0.0</pre>	Verifies the configuration.
Step 13	show ip bgp neighbor Example: <pre>Device# show ip bgp neighbor</pre>	<p>Verifies that NSF awareness (Graceful Restart) is enabled on the neighbor. If NSF awareness is enabled on the switch and the neighbor, this message appears: <i>Graceful Restart Capability: advertised and received</i></p> <p>If NSF awareness is enabled on the switch, but not on the neighbor, this message appears: <i>Graceful Restart Capability: advertised</i></p>
Step 14	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Managing Routing Policy Changes

To learn if a BGP peer supports the route refresh capability and to reset the BGP session:

Procedure

	Command or Action	Purpose
Step 1	show ip bgp neighbors Example: Device# show ip bgp neighbors	Displays whether a neighbor supports the route refresh capability. When supported, this message appears for the router: <i>Received route refresh capability from peer.</i>
Step 2	clear ip bgp { * address peer-group-name } Example: Device# clear ip bgp *	Resets the routing table on the specified connection. <ul style="list-style-type: none"> • Enter an asterisk (*) to specify that all connections be reset. • Enter an IP address to specify the connection to be reset. • Enter a peer group name to reset the peer group.
Step 3	clear ip bgp { * address peer-group-name } soft out Example: Device# clear ip bgp * soft out	(Optional) Performs an outbound soft reset to reset the inbound routing table on the specified connection. Use this command if route refresh is supported. <ul style="list-style-type: none"> • Enter an asterisk (*) to specify that all connections be reset. • Enter an IP address to specify the connection to be reset. • Enter a peer group name to reset the peer group.
Step 4	show ip bgp Example: Device# show ip bgp	Verifies the reset by checking information about the routing table and about BGP neighbors.
Step 5	show ip bgp neighbors Example: Device# show ip bgp neighbors	Verifies the reset by checking information about the routing table and about BGP neighbors.

Configuring BGP Decision Attributes

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system</i> Example: Device(config)# router bgp 4500	Enables a BGP routing process, assign it an AS number, and enter router configuration mode.
Step 4	bgp best-path as-path ignore Example: Device(config-router)# bgp bestpath as-path ignore	(Optional) Configures the router to ignore AS path length in selecting a route.
Step 5	neighbor {<i>ip-address</i> <i>peer-group-name</i>} next-hop-self Example: Device(config-router)# neighbor 10.108.1.1 next-hop-self	(Optional) Disables next-hop processing on BGP updates to a neighbor by entering a specific IP address to be used instead of the next-hop address.
Step 6	neighbor {<i>ip-address</i> <i>peer-group-name</i>} weight <i>weight</i> Example: Device(config-router)# neighbor 172.16.12.1 weight 50	(Optional) Assign a weight to a neighbor connection. Acceptable values are from 0 to 65535; the largest weight is the preferred route. Routes learned through another BGP peer have a default weight of 0; routes sourced by the local router have a default weight of 32768.
Step 7	default-metric <i>number</i> Example: Device(config-router)# default-metric 300	(Optional) Sets a MED metric to set preferred paths to external neighbors. All routes without a MED will also be set to this value. The range is 1 to 4294967295. The lowest value is the most desirable.
Step 8	bgp bestpath med missing-as-worst Example: Device(config-router)# bgp bestpath med missing-as-worst	(Optional) Configures the switch to consider a missing MED as having a value of infinity, making the path without a MED value the least desirable path.

	Command or Action	Purpose
Step 9	bgp always-compare med Example: Device(config-router)# bgp always-compare-med	(Optional) Configures the switch to compare MEDs for paths from neighbors in different autonomous systems. By default, MED comparison is only done among paths in the same AS.
Step 10	bgp bestpath med confed Example: Device(config-router)# bgp bestpath med confed	(Optional) Configures the switch to consider the MED in choosing a path from among those advertised by different subautonomous systems within a confederation.
Step 11	bgp deterministic med Example: Device(config-router)# bgp deterministic med	(Optional) Configures the switch to consider the MED variable when choosing among routes advertised by different peers in the same AS.
Step 12	bgp default local-preference value Example: Device(config-router)# bgp default local-preference 200	(Optional) Change the default local preference value. The range is 0 to 4294967295; the default value is 100. The highest local preference value is preferred.
Step 13	maximum-paths number Example: Device(config-router)# maximum-paths 8	(Optional) Configures the number of paths to be added to the IP routing table. The default is to only enter the best path in the routing table. The range is from 1 to 16. Having multiple paths allows load-balancing among the paths. (Although the switch software allows a maximum of 32 equal-cost routes, the switch hardware will never use more than 16 paths per route.)
Step 14	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 15	show ip bgp Example: Device# show ip bgp	Verifies the reset by checking information about the routing table and about BGP neighbors.
Step 16	show ip bgp neighbors Example: Device# show ip bgp neighbors	Verifies the reset by checking information about the routing table and about BGP neighbors.
Step 17	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	Device# <code>copy running-config startup-config</code>	

Configuring BGP Filtering with Route Maps

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Device(config)# <code>route-map set-peer-address permit 10</code>	Creates a route map, and enter route-map configuration mode.
Step 4	set ip next-hop <i>ip-address</i> [... <i>ip-address</i>] [peer-address] Example: Device(config)# <code>set ip next-hop 10.1.1.3</code>	(Optional) Sets a route map to disable next-hop processing <ul style="list-style-type: none"> • In an inbound route map, set the next hop of matching routes to be the neighbor peering address, overriding third-party next hops. • In an outbound route map of a BGP peer, set the next hop to the peering address of the local router, disabling the next-hop calculation.
Step 5	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 6	show route-map [<i>map-name</i>] Example: Device# <code>show route-map</code>	Displays all route maps configured or only the one specified to verify configuration.

	Command or Action	Purpose
Step 7	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring BGP Filtering by Neighbor

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system</i> Example: Device(config)# <code>router bgp 109</code>	Enables a BGP routing process, assign it an AS number, and enter router configuration mode.
Step 4	neighbor {<i>ip-address</i> <i>peer-group name</i>} distribute-list {<i>access-list-number</i> <i>name</i>} {<i>in</i> <i>out</i>} Example: Device(config-router)# <code>neighbor 172.16.4.1 distribute-list 39 in</code>	(Optional) Filters BGP routing updates to or from neighbors as specified in an access list. Note You can also use the neighbor prefix-list router configuration command to filter updates, but you cannot use both commands to configure the same BGP peer.
Step 5	neighbor {<i>ip-address</i> <i>peer-group name</i>} route-map <i>map-tag</i> {<i>in</i> <i>out</i>} Example: Device(config-router)# <code>neighbor 172.16.70.24 route-map internal-map in</code>	(Optional) Applies a route map to filter an incoming or outgoing route.
Step 6	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# end	
Step 7	show ip bgp neighbors Example: Device# show ip bgp neighbors	Verifies the configuration.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring BGP Filtering by Access Lists and Neighbors

Another method of filtering is to specify an access list filter on both incoming and outbound updates, based on the BGP autonomous system paths. Each filter is an access list based on regular expressions. To use this method, define an autonomous system path access list, and apply it to updates to and from particular neighbors.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip as-path access-list <i>access-list-number</i> {permit deny} <i>as-regular-expressions</i> Example: Device(config)# ip as-path access-list 1 deny _65535_	Defines a BGP-related access list.
Step 4	router bgp <i>autonomous-system</i> Example: Device(config)# router bgp 110	Enters BGP router configuration mode.

	Command or Action	Purpose
Step 5	neighbor { <i>ip-address</i> <i>peer-group name</i> } filter-list { <i>access-list-number</i> <i>name</i> } { in out weight weight } Example: <pre>Device(config-router)# neighbor 172.16.1.1 filter-list 1 out</pre>	Establishes a BGP filter based on an access list.
Step 6	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 7	show ip bgp neighbors [<i>paths regular-expression</i>] Example: <pre>Device# show ip bgp neighbors</pre>	Verifies the configuration.
Step 8	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Prefix Lists for BGP Filtering

You do not need to specify a sequence number when removing a configuration entry. **Show** commands include the sequence numbers in their output.

Before using a prefix list in a command, you must set up the prefix list.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] deny permit <i>network/len</i> [ge <i>ge-value</i>] [le <i>le-value</i>]</p> <p>Example:</p> <pre>Device(config)# ip prefix-list BLUE permit 172.16.1.0/24</pre>	<p>Creates a prefix list with an optional sequence number to deny or permit access for matching conditions. You must enter at least one permit or deny clause.</p> <ul style="list-style-type: none"> <i>network/len</i> is the network number and length (in bits) of the network mask. (Optional) ge and le values specify the range of the prefix length to be matched. The specified <i>ge-value</i> and <i>le-value</i> must satisfy this condition: $len < ge-value < le-value < 32$
Step 4	<p>ip prefix-list <i>list-name</i> seq <i>seq-value</i> deny permit <i>network/len</i> [ge <i>ge-value</i>] [le <i>le-value</i>]</p> <p>Example:</p> <pre>Device(config)# ip prefix-list BLUE seq 10 permit 172.24.1.0/24</pre>	(Optional) Adds an entry to a prefix list, and assign a sequence number to the entry.
Step 5	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	<p>show ip prefix list [detail summary] <i>name</i> [<i>network/len</i>] [seq <i>seq-num</i>] [longer] [first-match]</p> <p>Example:</p> <pre>Device# show ip prefix list summary test</pre>	Verifies the configuration by displaying information about a prefix list or prefix list entries.
Step 7	<p>copy running-config startup-config</p> <p>Example:</p> <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring BGP Community Filtering

By default, no COMMUNITIES attribute is sent to a neighbor. You can specify that the COMMUNITIES attribute be sent to the neighbor at an IP address by using the **neighbor send-community** router configuration command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip community-list** *community-list-number* {**permit** | **deny**} *community-number*

4. **router bgp** *autonomous-system*
5. **neighbor** {*ip-address* | *peer-group name*} **send-community**
6. **set comm-list** *list-num* **delete**
7. **exit**
8. **ip bgp-community new-format**
9. **end**
10. **show ip bgp community**
11. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip community-list <i>community-list-number</i> { permit deny } <i>community-number</i> Example: Device(config)# ip community-list 1 permit 50000:10	Creates a community list, and assigns it a number. <ul style="list-style-type: none"> • The <i>community-list-number</i> is an integer from 1 to 99 that identifies one or more permit or deny groups of communities. • The <i>community-number</i> is the number configured by a set community route-map configuration command.
Step 4	router bgp <i>autonomous-system</i> Example: Device(config)# router bgp 108	Enters BGP router configuration mode.
Step 5	neighbor { <i>ip-address</i> <i>peer-group name</i> } send-community Example: Device(config-router)# neighbor 172.16.70.23 send-community	Specifies that the COMMUNITIES attribute be sent to the neighbor at this IP address.
Step 6	set comm-list <i>list-num</i> delete Example: Device(config-router)# set comm-list 500 delete	(Optional) Removes communities from the community attribute of an inbound or outbound update that match a standard or extended community list specified by a route map.

	Command or Action	Purpose
Step 7	exit Example: Device(config-router)# end	Returns to global configuration mode.
Step 8	ip bgp-community new-format Example: Device(config)# ip bgp-community new format	(Optional) Displays and parses BGP communities in the format AA:NN. A BGP community is displayed in a two-part format 2 bytes long. The Cisco default community format is in the format NNAA. In the most recent RFC for BGP, a community takes the form AA:NN, where the first part is the AS number and the second part is a 2-byte number.
Step 9	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 10	show ip bgp community Example: Device# show ip bgp community	Verifies the configuration.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring BGP Neighbors and Peer Groups

To assign configuration options to an individual neighbor, specify any of these router configuration commands by using the neighbor IP address. To assign the options to a peer group, specify any of the commands by using the peer group name. You can disable a BGP peer or peer group without removing all the configuration information by using the **neighbor shutdown** router configuration command.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system</i>	Enters BGP router configuration mode.
Step 4	neighbor <i>peer-group-name</i> peer-group	Creates a BGP peer group.
Step 5	neighbor <i>ip-address</i> peer-group <i>peer-group-name</i>	Makes a BGP neighbor a member of the peer group.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>number</i>	Specifies a BGP neighbor. If a peer group is not configured with a remote-as <i>number</i> , use this command to create peer groups containing EBGP neighbors. The range is 1 to 65535.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> } description <i>text</i>	(Optional) Associates a description with a neighbor.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> } default-originate [route-map <i>map-name</i>]	(Optional) Allows a BGP speaker (the local router) to send the default route 0.0.0.0 to a neighbor for use as a default route.
Step 9	neighbor { <i>ip-address</i> <i>peer-group-name</i> } send-community	(Optional) Specifies that the COMMUNITIES attribute be sent to the neighbor at this IP address.
Step 10	neighbor { <i>ip-address</i> <i>peer-group-name</i> } update-source <i>interface</i>	(Optional) Allows internal BGP sessions to use any operational interface for TCP connections.
Step 11	neighbor { <i>ip-address</i> <i>peer-group-name</i> } ebgp-multihop	(Optional) Allows BGP sessions, even when the neighbor is not on a directly connected segment. The multihop session is not established if the only route to the multihop peer's address is the default route (0.0.0.0).
Step 12	neighbor { <i>ip-address</i> <i>peer-group-name</i> } local-as <i>number</i>	(Optional) Specifies an AS number to use as the local AS. The range is 1 to 65535.
Step 13	neighbor { <i>ip-address</i> <i>peer-group-name</i> } advertisement-interval <i>seconds</i>	(Optional) Sets the minimum interval between sending BGP routing updates.
Step 14	neighbor { <i>ip-address</i> <i>peer-group-name</i> } maximum-prefix <i>maximum</i> [<i>threshold</i>]	(Optional) Controls how many prefixes can be received from a neighbor. The range is 1 to 4294967295. The <i>threshold</i> (optional) is the percentage of maximum at which a warning message is generated. The default is 75 percent.
Step 15	neighbor { <i>ip-address</i> <i>peer-group-name</i> } next-hop-self	(Optional) Disables next-hop processing on the BGP updates to a neighbor.
Step 16	neighbor { <i>ip-address</i> <i>peer-group-name</i> } password <i>string</i>	(Optional) Sets MD5 authentication on a TCP connection to a BGP peer. The same password must be configured on

	Command or Action	Purpose
		both BGP peers, or the connection between them is not made.
Step 17	neighbor { <i>ip-address</i> <i>peer-group-name</i> } route-map <i>map-name</i> { in out }	(Optional) Applies a route map to incoming or outgoing routes.
Step 18	neighbor { <i>ip-address</i> <i>peer-group-name</i> } send-community	(Optional) Specifies that the COMMUNITIES attribute be sent to the neighbor at this IP address.
Step 19	neighbor { <i>ip-address</i> <i>peer-group-name</i> } timers <i>keepalive holdtime</i>	(Optional) Sets timers for the neighbor or peer group. <ul style="list-style-type: none"> The <i>keepalive</i> interval is the time within which keepalive messages are sent to peers. The range is 1 to 4294967295 seconds; the default is 60. The <i>holdtime</i> is the interval after which a peer is declared inactive after not receiving a keepalive message from it. The range is 1 to 4294967295 seconds; the default is 180.
Step 20	neighbor { <i>ip-address</i> <i>peer-group-name</i> } weight <i>weight</i>	(Optional) Specifies a weight for all routes from a neighbor.
Step 21	neighbor { <i>ip-address</i> <i>peer-group-name</i> } distribute-list { <i>access-list-number</i> <i>name</i> } { in out }	(Optional) Filter BGP routing updates to or from neighbors, as specified in an access list.
Step 22	neighbor { <i>ip-address</i> <i>peer-group-name</i> } filter-list <i>access-list-number</i> { in out weight <i>weight</i> }	(Optional) Establish a BGP filter.
Step 23	neighbor { <i>ip-address</i> <i>peer-group-name</i> } version <i>value</i>	(Optional) Specifies the BGP version to use when communicating with a neighbor.
Step 24	neighbor { <i>ip-address</i> <i>peer-group-name</i> } soft-reconfiguration inbound	(Optional) Configures the software to start storing received updates.
Step 25	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 26	show ip bgp neighbors	Verifies the configuration.
Step 27	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Aggregate Addresses in a Routing Table

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system</i> Example: Device(config)# router bgp 106	Enters BGP router configuration mode.
Step 4	aggregate-address <i>address mask</i> Example: Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0	Creates an aggregate entry in the BGP routing table. The aggregate route is advertised as coming from the AS, and the atomic aggregate attribute is set to indicate that information might be missing.
Step 5	aggregate-address <i>address mask as-set</i> Example: Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 as-set	(Optional) Generates AS set path information. This command creates an aggregate entry following the same rules as the previous command, but the advertised path will be an AS_SET consisting of all elements contained in all paths. Do not use this keyword when aggregating many paths because this route must be continually withdrawn and updated.
Step 6	aggregate-address <i>address-mask summary-only</i> Example: Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 summary-only	(Optional) Advertises summary addresses only.
Step 7	aggregate-address <i>address mask suppress-map map-name</i> Example: Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 suppress-map map1	(Optional) Suppresses selected, more specific routes.

	Command or Action	Purpose
Step 8	aggregate-address <i>address mask</i> advertise-map <i>map-name</i> Example: <pre>Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 advertise-map map2</pre>	(Optional) Generates an aggregate based on conditions specified by the route map.
Step 9	aggregate-address <i>address mask</i> attribute-map <i>map-name</i> Example: <pre>Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 attribute-map map3</pre>	(Optional) Generates an aggregate with attributes specified in the route map.
Step 10	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 11	show ip bgp neighbors [<i>advertised-routes</i>] Example: <pre>Device# show ip bgp neighbors</pre>	Verifies the configuration.
Step 12	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Routing Domain Confederations

You must specify a confederation identifier that acts as the autonomous system number for the group of autonomous systems.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	router bgp <i>autonomous-system</i> Example: Device(config)# <code>router bgp 100</code>	Enters BGP router configuration mode.
Step 4	bgp confederation identifier <i>autonomous-system</i> Example: Device(config)# <code>bgp confederation identifier 50007</code>	Configures a BGP confederation identifier.
Step 5	bgp confederation peers <i>autonomous-system</i> [<i>autonomous-system ...</i>] Example: Device(config)# <code>bgp confederation peers 51000 51001 51002</code>	Specifies the autonomous systems that belong to the confederation and that will be treated as special EBGP peers.
Step 6	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.
Step 7	show ip bgp neighbor Example: Device# <code>show ip bgp neighbor</code>	Verifies the configuration.
Step 8	show ip bgp network Example: Device# <code>show ip bgp network</code>	Verifies the configuration.
Step 9	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring BGP Route Reflectors

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system</i> Example: Device(config)# router bgp 101	Enters BGP router configuration mode.
Step 4	neighbor { <i>ip-address</i> <i>peer-group-name</i> } route-reflector-client Example: Device(config-router)# neighbor 172.16.70.24 route-reflector-client	Configures the local router as a BGP route reflector and the specified neighbor as a client.
Step 5	bgp cluster-id <i>cluster-id</i> Example: Device(config-router)# bgp cluster-id 10.0.1.2	(Optional) Configures the cluster ID if the cluster has more than one route reflector.
Step 6	no bgp client-to-client reflection Example: Device(config-router)# no bgp client-to-client reflection	(Optional) Disables client-to-client route reflection. By default, the routes from a route reflector client are reflected to other clients. However, if the clients are fully meshed, the route reflector does not need to reflect routes to clients.
Step 7	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 8	show ip bgp Example: Device# show ip bgp	Verifies the configuration. Displays the originator ID and the cluster-list attributes.

	Command or Action	Purpose
Step 9	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring Route Dampening

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system</i> Example: Device(config)# <code>router bgp 100</code>	Enters BGP router configuration mode.
Step 4	bgp dampening Example: Device(config-router)# <code>bgp dampening</code>	Enables BGP route dampening.
Step 5	bgp dampening <i>half-life reuse suppress max-suppress</i> [route-map <i>map</i>] Example: Device(config-router)# <code>bgp dampening 30 1500 10000 120</code>	(Optional) Changes the default values of route dampening factors.
Step 6	end Example: Device(config)# <code>end</code>	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	show ip bgp flap-statistics [<i>{regexp regexp}</i> <i>{filter-list list}</i> <i>{address mask [longer-prefix]}</i>] Example: Device# show ip bgp flap-statistics	(Optional) Monitors the flaps of all paths that are flapping. The statistics are deleted when the route is not suppressed and is stable.
Step 8	show ip bgp dampened-paths Example: Device# show ip bgp dampened-paths	(Optional) Displays the dampened routes, including the time remaining before they are suppressed.
Step 9	clear ip bgp flap-statistics [<i>{regexp regexp}</i> <i>{filter-list list}</i> <i>{address mask [longer-prefix]}</i>] Example: Device# clear ip bgp flap-statistics	(Optional) Clears BGP flap statistics to make it less likely that a route will be dampened.
Step 10	clear ip bgp dampening Example: Device# clear ip bgp dampening	(Optional) Clears route dampening information, and unsuppress the suppressed routes.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Conditionally Injecting BGP Routes

Use this task to inject more specific prefixes into a BGP routing table over less specific prefixes that were selected through normal route aggregation. These more specific prefixes can be used to provide a finer granularity of traffic engineering or administrative control than is possible with aggregated routes.

Before you begin

This task assumes that the IGP is already configured for the BGP peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **bgp inject-map** *inject-map-name* **exist-map** *exist-map-name* [*copy-attributes*]
5. **exit**
6. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]

7. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
8. **match ip route-source** {*access-list-number* | *access-list-name*} [*access-list-number...* | *access-list-name...*]
9. **exit**
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **set ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
12. **set community** {*community-number* [**additive**] [*well-known-community*] | **none**}
13. **exit**
14. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} [**ge** *ge-value*] [**le** *le-value*]
15. Repeat Step 14 for every prefix list to be created.
16. **exit**
17. **show ip bgp injected-paths**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 40000	Enters router configuration mode for the specified routing process.
Step 4	bgp inject-map <i>inject-map-name</i> exist-map <i>exist-map-name</i> [copy-attributes] Example: Device(config-router)# bgp inject-map ORIGINATE exist-map LEARNED_PATH	Specifies the inject map and the exist map for conditional route injection. <ul style="list-style-type: none"> • Use the copy-attributes keyword to specify that the injected route inherit the attributes of the aggregate route.
Step 5	exit Example: Device(config-router)# exit	Exits router configuration mode and enters global configuration mode.
Step 6	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example:	Configures a route map and enters route map configuration mode.

	Command or Action	Purpose
	Device(config)# route-map LEARNED_PATH permit 10	
Step 7	<p>match ip address {<i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]}</p> <p>Example:</p> <pre>Device(config-route-map)# match ip address prefix-list SOURCE</pre>	<p>Specifies the aggregate route to which a more specific route will be injected.</p> <ul style="list-style-type: none"> In this example, the prefix list named SOURCE is used to redistribute the source of the route.
Step 8	<p>match ip route-source {<i>access-list-number</i> <i>access-list-name</i>} [<i>access-list-number...</i> <i>access-list-name...</i>]</p> <p>Example:</p> <pre>Device(config-route-map)# match ip route-source prefix-list ROUTE_SOURCE</pre>	<p>Specifies the match conditions for redistributing the source of the route.</p> <ul style="list-style-type: none"> In this example, the prefix list named ROUTE_SOURCE is used to redistribute the source of the route. <p>Note The route source is the neighbor address that is configured with the neighbor remote-as command. The tracked prefix must come from this neighbor in order for conditional route injection to occur.</p>
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config-route-map)# exit</pre>	Exits route map configuration mode and enters global configuration mode.
Step 10	<p>route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map ORIGINATE permit 10</pre>	Configures a route map and enters route map configuration mode.
Step 11	<p>set ip address {<i>access-list-number</i> [<i>access-list-number...</i> <i>access-list-name...</i>] <i>access-list-name</i> [<i>access-list-number...</i> <i>access-list-name</i>] prefix-list <i>prefix-list-name</i> [<i>prefix-list-name...</i>]}</p> <p>Example:</p> <pre>Device(config-route-map)# set ip address prefix-list ORIGINATED_ROUTES</pre>	<p>Specifies the routes to be injected.</p> <ul style="list-style-type: none"> In this example, the prefix list named originated_routes is used to redistribute the source of the route.
Step 12	<p>set community {<i>community-number</i> [additive] [<i>well-known-community</i>] none}</p> <p>Example:</p> <pre>Device(config-route-map)# set community 14616:555 additive</pre>	Sets the BGP community attribute of the injected route.

	Command or Action	Purpose
Step 13	exit Example: <pre>Device(config-route-map)# exit</pre>	Exits route map configuration mode and enters global configuration mode.
Step 14	ip prefix-list <i>list-name</i> [seq <i>seq-value</i>] { deny <i>network/length</i> permit <i>network/length</i> } [ge <i>ge-value</i>] [le <i>le-value</i>] Example: <pre>Device(config)# ip prefix-list SOURCE permit 10.1.1.0/24</pre>	Configures a prefix list. <ul style="list-style-type: none"> In this example, the prefix list named SOURCE is configured to permit routes from network 10.1.1.0/24.
Step 15	Repeat Step 14 for every prefix list to be created.	--
Step 16	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.
Step 17	show ip bgp injected-paths Example: <pre>Device# show ip bgp injected-paths</pre>	(Optional) Displays information about injected paths.

Configuring Peer Session Templates

Use the following tasks to create and configure a peer session template:

Configuring a Basic Peer Session Template

Perform this task to create a basic peer session template with general BGP routing session commands that can be applied to many neighbors using one of the next two tasks.



Note The commands in Step 5 and 6 are optional and could be replaced with any supported general session commands.



Note The following restrictions apply to the peer session templates:

- A peer session template can directly inherit only one session template, and each inherited session template can also contain one indirectly inherited session template. So, a neighbor or neighbor group can be configured with only one directly applied peer session template and seven additional indirectly inherited peer session templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

To configure a basic peer session template, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **remote-as** *autonomous-system-number*
6. **timers** *keepalive-interval hold-time*
7. **end**
8. **show ip bgp template peer-session** [*session-template-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 101	Enters router configuration mode and creates a BGP routing process.
Step 4	template peer-session <i>session-template-name</i> Example: Device(config-router)# template peer-session INTERNAL-BGP	Enters session-template configuration mode and creates a peer session template.

	Command or Action	Purpose
Step 5	remote-as <i>autonomous-system-number</i> Example: Device(config-router-stmp)# remote-as 202	(Optional) Configures peering with a remote neighbor in the specified autonomous system. Note Any supported general session command can be used here. For a list of the supported commands, see the “Restrictions” section.
Step 6	timers <i>keepalive-interval hold-time</i> Example: Device(config-router-stmp)# timers 30 300	(Optional) Configures BGP keepalive and hold timers. The hold time must be at least twice the keepalive time. Note Any supported general session command can be used here. For a list of the supported commands, see the “Restrictions” section.
Step 7	end Example: Device(config-router)# end	Exits session-template configuration mode and returns to privileged EXEC mode.
Step 8	show ip bgp template peer-session <i>[session-template-name]</i> Example: Device# show ip bgp template peer-session	Displays locally configured peer session templates. The output can be filtered to display a single peer policy template with the <i>session-template-name</i> argument. This command also supports all standard output modifiers.

Configuring Peer Session Template Inheritance with the `inherit peer-session` Command

This task configures peer session template inheritance with the **inherit peer-session** command. It creates and configures a peer session template and allows it to inherit a configuration from another peer session template.



Note The commands in Steps 5 and 6 are optional and could be replaced with any supported general session commands.

To configure peer session template inheritance, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-session** *session-template-name*
5. **description** *text-string*
6. **update-source** *interface-type interface-number*
7. **inherit peer-session** *session-template-name*
8. **end**

9. show ip bgp template peer-session [session-template-name]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 101	Enters router configuration mode and creates a BGP routing process.
Step 4	template peer-session <i>session-template-name</i> Example: Device(config-router)# template peer-session CORE1	Enter session-template configuration mode and creates a peer session template.
Step 5	description <i>text-string</i> Example: Device(config-router-stmp)# description CORE-123	(Optional) Configures a description. The text string can be up to 80 characters. Note Any supported general session command can be used here. For a list of the supported commands, see the “Restrictions” section.
Step 6	update-source <i>interface-type interface-number</i> Example: Device(config-router-stmp)# update-source loopback 1	(Optional) Configures a router to select a specific source or interface to receive routing table updates. The example uses a loopback interface. The advantage to this configuration is that the loopback interface is not as susceptible to the effects of a flapping interface. Note Any supported general session command can be used here. For a list of the supported commands, see the “Restrictions” section.
Step 7	inherit peer-session <i>session-template-name</i> Example: Device(config-router-stmp)# inherit peer-session INTERNAL-BGP	Configures this peer session template to inherit the configuration of another peer session template. The example configures this peer session template to inherit the configuration from INTERNAL-BGP. This template can be applied to a neighbor, and the configuration INTERNAL-BGP will be applied indirectly. No additional

	Command or Action	Purpose
		peer session templates can be directly applied. However, the directly inherited template can contain up to seven indirectly inherited peer session templates.
Step 8	end Example: Device(config-router)# end	Exits session-template configuration mode and enters privileged EXEC mode.
Step 9	show ip bgp template peer-session [session-template-name] Example: Device# show ip bgp template peer-session	Displays locally configured peer session templates. The output can be filtered to display a single peer policy template with the optional <i>session-template-name</i> argument. This command also supports all standard output modifiers.

Configuring Peer Session Template Inheritance with the `neighbor inherit peer-session` Command

This task configures a device to send a peer session template to a neighbor to inherit the configuration from the specified peer session template with the **neighbor inherit peer-session** command. Use the following steps to send a peer session template configuration to a neighbor to inherit.

To configure peer session template inheritance, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **neighbor** *ip-address* **inherit peer-session** *session-template-name*
6. **end**
7. **show ip bgp template peer-session** [*session-template-name*]

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 101	Enters router configuration mode and creates a BGP routing process.
Step 4	neighbor ip-address remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 172.16.0.1 remote-as 202	Configures a peering session with the specified neighbor. The explicit remote-as statement is required for the neighbor inherit statement in Step 5 to work. If a peering is not configured, the specified neighbor in Step 5 will not accept the session template.
Step 5	neighbor ip-address inherit peer-session <i>session-template-name</i> Example: Device(config-router)# neighbor 172.16.0.1 inherit peer-session CORE1	Sends a peer session template to a neighbor so that the neighbor can inherit the configuration. The example configures a device to send the peer session template named CORE1 to the 172.16.0.1 neighbor to inherit. This template can be applied to a neighbor, and if another peer session template is indirectly inherited in CORE1, the indirectly inherited configuration will also be applied. No additional peer session templates can be directly applied. However, the directly inherited template can also inherit up to seven additional indirectly inherited peer session templates.
Step 6	end Example: Device(config-router)# end	Exits router configuration mode and enters privileged EXEC mode.
Step 7	show ip bgp template peer-session <i>[session-template-name]</i> Example: Device# show ip bgp template peer-session	Displays locally configured peer session templates. The output can be filtered to display a single peer policy template with the optional <i>session-template-name</i> argument. This command also supports all standard output modifiers.

Configuring Peer Policy Templates

Use the following tasks to create and configure a peer policy template:

Configuring Basic Peer Policy Templates

Perform this task to create a basic peer policy template with BGP policy configuration commands that can be applied to many neighbors using one of the next two tasks.



Note The commands in Steps 5 through 7 are optional and could be replaced with any supported BGP policy configuration commands.



Note The following restrictions apply to the peer policy templates:

- A peer policy template can directly or indirectly inherit up to eight peer policy templates.
- A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

To configure basic peer policy templates, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **template peer-policy** *policy-template-name*
5. **maximum-prefix** *prefix-limit* [*threshold*] [**restart** *restart-interval* | **warning-only**]
6. **weight** *weight-value*
7. **prefix-list** *prefix-list-name* {**in** | **out**}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	template peer-policy <i>policy-template-name</i> Example:	Enters policy-template configuration mode and creates a peer policy template.

	Command or Action	Purpose
	Device(config-router)# <code>template peer-policy GLOBAL</code>	
Step 5	<p>maximum-prefix <i>prefix-limit</i> [<i>threshold</i>] [restart <i>restart-interval</i> warning-only]</p> <p>Example:</p> <pre>Device(config-router-ptmp)# maximum-prefix 10000</pre>	<p>(Optional) Configures the maximum number of prefixes that a neighbor accept from this peer.</p> <p>Note Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the “Peer Policy Templates” section.</p>
Step 6	<p>weight <i>weight-value</i></p> <p>Example:</p> <pre>Device(config-router-ptmp)# weight 300</pre>	<p>(Optional) Sets the default weight for routes that are sent from this neighbor.</p> <p>Note Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the “Peer Policy Templates” section.</p>
Step 7	<p>prefix-list <i>prefix-list-name</i> {in out}</p> <p>Example:</p> <pre>Device(config-router-ptmp)# prefix-list NO-MARKETING in</pre>	<p>(Optional) Filters prefixes that are received by the router or sent from the router.</p> <p>The prefix list in the example filters inbound internal addresses.</p> <p>Note Any supported BGP policy configuration command can be used here. For a list of the supported commands, see the “Peer Policy Templates” section.</p>
Step 8	<p>end</p> <p>Example:</p> <pre>Device(config-router-ptmp)# end</pre>	<p>Exits policy-template configuration mode and returns to privileged EXEC mode.</p>

Configuring Peer Policy Template Inheritance with the `inherit peer-policy` Command

This task configures peer policy template inheritance using the `inherit peer-policy` command. It creates and configure a peer policy template and allows it to inherit a configuration from another peer policy template.



Note The commands in Steps 5 and 6 are optional and could be replaced with any supported BGP policy configuration commands.

To configure peer policy template inheritance, perform this procedure:

SUMMARY STEPS

1. `enable`
2. `configure terminal`

3. `router bgp` *autonomous-system-number*
4. `template peer-policy` *policy-template-name*
5. `route-map` *map-name* {**in**|**out**}
6. `inherit peer-policy` *policy-template-name* *sequence-number*
7. `end`
8. `show ip bgp template peer-policy` [*policy-template-name*[**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	template peer-policy <i>policy-template-name</i> Example: Device(config-router)# template peer-policy NETWORK1	Enter policy-template configuration mode and creates a peer policy template.
Step 5	route-map <i>map-name</i> { in out } Example: Device(config-router-ptmp)# route-map ROUTE in	(Optional) Applies the specified route map to inbound or outbound routes. Note Any supported BGP policy configuration command can be used here.
Step 6	inherit peer-policy <i>policy-template-name</i> <i>sequence-number</i> Example: Device(config-router-ptmp)# inherit peer-policy GLOBAL 10	Configures the peer policy template to inherit the configuration of another peer policy template. <ul style="list-style-type: none"> • The <i>sequence-number</i> argument sets the order in which the peer policy template is evaluated. Like a route map sequence number, the lowest sequence number is evaluated first. • The example configures this peer policy template to inherit the configuration from GLOBAL. If the template created in these steps is applied to a neighbor, the configuration GLOBAL will also be inherited and applied indirectly. Up to six additional peer policy

	Command or Action	Purpose
		<p>templates can be indirectly inherited from GLOBAL for a total of eight directly applied and indirectly inherited peer policy templates.</p> <ul style="list-style-type: none"> This template in the example will be evaluated first if no other templates are configured with a lower sequence number.
Step 7	<p>end</p> <p>Example:</p> <pre>Device(config-router-ptmp)# end</pre>	Exits policy-template configuration mode and returns to privileged EXEC mode.
Step 8	<p>show ip bgp template peer-policy [<i>policy-template-name</i>[detail]]</p> <p>Example:</p> <pre>Device# show ip bgp template peer-policy NETWORK1 detail</pre>	<p>Displays locally configured peer policy templates.</p> <ul style="list-style-type: none"> The output can be filtered to display a single peer policy template with the <i>policy-template-name</i> argument. This command also supports all standard output modifiers. Use the detail keyword to display detailed policy information.

Examples

The following sample output of the **show ip bgp template peer-policy** command with the **detail** keyword displays details of the policy named NETWORK1. The output in this example shows that the GLOBAL template was inherited. Details of route map and prefix list configurations are also displayed.

```
Device# show ip bgp template peer-policy NETWORK1 detail
Template:NETWORK1, index:2.
Local policies:0x1, Inherited polices:0x80840
This template inherits:
  GLOBAL, index:1, seq_no:10, flags:0x1
Locally configured policies:
  route-map ROUTE in
Inherited policies:
  prefix-list NO-MARKETING in
  weight 300
  maximum-prefix 10000
Template:NETWORK1 <detail>
Locally configured policies:
  route-map ROUTE in
route-map ROUTE, permit, sequence 10
Match clauses:
  ip address prefix-lists: DEFAULT
ip prefix-list DEFAULT: 1 entries
  seq 5 permit 10.1.1.0/24
Set clauses:
  Policy routing matches: 0 packets, 0 bytes
Inherited policies:
  prefix-list NO-MARKETING in
```

```
ip prefix-list NO-MARKETING: 1 entries
seq 5 deny 10.2.2.0/24
```

Configuring Peer Policy Template Inheritance with the `neighbor inherit peer-policy` Command

This task configures a device to send a peer policy template to a neighbor to inherit using the **neighbor inherit peer-policy** command. Perform the following steps to send a peer policy template configuration to a neighbor to inherit.

When BGP neighbors use multiple levels of peer templates, it can be difficult to determine which policies are applied to the neighbor. The **policy** and **detail** keywords of the **show ip bgp neighbors** command display the inherited policies and policies that are configured directly on the specified neighbor.

To configure peer policy template, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
5. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
6. **neighbor** *ip-address* **inherit peer-policy** *policy-template-name*
7. **end**
8. **show ip bgp neighbors** [*ip-address*[**policy** [**detail**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 45000	Enters router configuration mode and creates a BGP routing process.
Step 4	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 192.168.1.2 remote-as 40000	Configures a peering session with the specified neighbor. The explicit remote-as statement is required for the neighbor inherit statement in Step 6 to work. If a peering is not configured, the specified neighbor in Step 6 will not accept the session template.

	Command or Action	Purpose
Step 5	address-family ipv4 [multicast unicast vrf vrf-name] Example: Device(config-router)# address-family ipv4 unicast	Enters address family configuration mode to configure a neighbor to accept address family-specific command configurations.
Step 6	neighbor ip-address inherit peer-policy <i>policy-template-name</i> Example: Device(config-router-af)# neighbor 192.168.1.2 inherit peer-policy GLOBAL	<p>Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.</p> <p>The example configures a router to send the peer policy template that is named GLOBAL to the 192.168.1.2 neighbor to inherit. This template can be applied to a neighbor, and if another peer policy template is indirectly inherited from GLOBAL, the indirectly inherited configuration will also be applied. Up to seven additional peer policy templates can be indirectly inherited from GLOBAL.</p>
Step 7	end Example: Device(config-router-af)# end	Exits address family configuration mode and returns to privileged EXEC mode.
Step 8	show ip bgp neighbors [<i>ip-address</i> [policy [detail]]] Example: Device# show ip bgp neighbors 192.168.1.2 policy	<p>Displays locally configured peer policy templates.</p> <ul style="list-style-type: none"> • The output can be filtered to display a single peer policy template with the <i>policy-template-name</i> argument. This command also supports all standard output modifiers. • Use the policy keyword to display the policies that are applied to this neighbor per address family. • Use the detail keyword to display detailed policy information.

Examples

The following sample output shows the policies that are applied to the neighbor at 192.168.1.2. The output displays both inherited policies and policies that are configured on the neighbor device. Inherited policies are policies that the neighbor inherits from a peer-group or a peer-policy template.

```
Device# show ip bgp neighbors 192.168.1.2 policy
Neighbor: 192.168.1.2, Address-Family: IPv4 Unicast
Locally configured policies:
  route-map ROUTE in
Inherited policies:
  prefix-list NO-MARKETING in
  route-map ROUTE in
  weight 300
  maximum-prefix 10000
```

Configuration Examples for BGP

Example: Configuring Conditional BGP Route Injection

The following sample output is similar to the output that will be displayed when the **show ip bgp injected-paths** command is entered:

```
Device# show ip bgp injected-paths

BGP table version is 11, local router ID is 10.0.0.1
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
*> 172.16.0.0       10.0.0.2              0 ?
*> 172.17.0.0/16   10.0.0.2              0 ?
```

Example: Configuring Peer Session Templates

The following example creates a peer session template named INTERNAL-BGP in session-template configuration mode:

```
router bgp 45000
 template peer-session INTERNAL-BGP
 remote-as 50000
 timers 30 300
 exit-peer-session
```

The following example creates a peer session template named CORE1. This example inherits the configuration of the peer session template named INTERNAL-BGP.

```
router bgp 45000
 template peer-session CORE1
 description CORE-123
 update-source loopback 1
 inherit peer-session INTERNAL-BGP
 exit-peer-session
```

The following example configures the 192.168.3.2 neighbor to inherit the CORE1 peer session template. The 192.168.3.2 neighbor will also indirectly inherit the configuration from the peer session template named INTERNAL-BGP. The explicit **remote-as** statement is required for the neighbor inherit statement to work. If a peering is not configured, the specified neighbor will not accept the session template.

```
router bgp 45000
 neighbor 192.168.3.2 remote-as 50000
 neighbor 192.168.3.2 inherit peer-session CORE1
```

Examples: Configuring Peer Policy Templates

The following example creates a peer policy template named GLOBAL and enters policy-template configuration mode:

```
router bgp 45000
  template peer-policy GLOBAL
  weight 1000
  maximum-prefix 5000
  prefix-list NO_SALES in
  exit-peer-policy
```

The following example creates a peer policy template named PRIMARY-IN and enters policy-template configuration mode:

```
router bgp 45000
  template peer-policy PRIMARY-IN
  prefix-list ALLOW-PRIMARY-A in
  route-map SET-LOCAL in
  weight 2345
  default-originate
  exit-peer-policy
```

The following example creates a peer policy template named CUSTOMER-A. This peer policy template is configured to inherit the configuration from the peer policy templates named PRIMARY-IN and GLOBAL.

```
router bgp 45000
  template peer-policy CUSTOMER-A
  route-map SET-COMMUNITY in
  filter-list 20 in
  inherit peer-policy PRIMARY-IN 20
  inherit peer-policy GLOBAL 10
  exit-peer-policy
```

The following example configures the 192.168.2.2 neighbor in address family mode to inherit the peer policy template named CUSTOMER-A. Assuming this example is a continuation of the example above, because the peer policy template named CUSTOMER-A above inherited the configuration from the templates named PRIMARY-IN and GLOBAL, the 192.168.2.2 neighbor will also indirectly inherit the peer policy templates named PRIMARY-IN and GLOBAL.

```
router bgp 45000
  neighbor 192.168.2.2 remote-as 50000
  address-family ipv4 unicast
    neighbor 192.168.2.2 inherit peer-policy CUSTOMER-A
  end
```

Monitoring and Maintaining BGP

You can remove all contents of a particular cache, table, or database. This might be necessary when the contents of the particular structure have become or are suspected to be invalid.

You can display specific statistics, such as the contents of BGP routing tables, caches, and databases. You can use the information to get resource utilization and solve network problems. You can also display information about node reachability and discover the routing path your device's packets are taking through the network.

The table given below lists the privileged EXEC commands for clearing and displaying BGP.

Table 8: IP BGP Clear and Show Commands

clear ip bgp address	Resets a particular BGP connection.
-----------------------------	-------------------------------------

clear ip bgp *	Resets all BGP connections.
clear ip bgp peer-group <i>tag</i>	Removes all members of a BGP peer group.
show ip bgp <i>prefix</i>	Displays peer groups and peers not in peer groups to which has been advertised. Also displays prefix attributes such as hop and the local prefix.
show ip bgp cidr-only	Displays all BGP routes that contain subnet and supernet net masks.
show ip bgp community [<i>community-number</i>] [exact]	Displays routes that belong to the specified communities.
show ip bgp community-list <i>community-list-number</i> [exact-match]	Displays routes that are permitted by the community list.
show ip bgp filter-list <i>access-list-number</i>	Displays routes that are matched by the specified AS path access list.
show ip bgp inconsistent-as	Displays the routes with inconsistent originating autonomous system numbers.
show ip bgp regexp <i>regular-expression</i>	Displays the routes that have an AS path that matches the specified regular expression entered on the command line.
show ip bgp	Displays the contents of the BGP routing table.
show ip bgp neighbors [<i>address</i>]	Displays detailed information on the BGP and TCP connections to individual neighbors.
show ip bgp neighbors [<i>address</i>] [advertised-routes dampened-routes flap-statistics paths <i>regular-expression</i> received-routes routes]	Displays routes learned from a particular BGP neighbor.
show ip bgp paths	Displays all BGP paths in the database.
show ip bgp peer-group [<i>tag</i>] [summary]	Displays information about BGP peer groups.
show ip bgp summary	Displays the status of all BGP connections.

The **bgp log-neighbor changes** command is enabled by default. It allows to log messages that are generated when a BGP neighbor resets, comes up, or goes down.