MPLS Virtual Private Networks (VPNs)

The IP Virtual Private Network (VPN) feature for Multiprotocol Label Switching (MPLS) allows a Cisco IOS network to deploy scalable IPv4 Layer 3 VPN backbone services. An IP VPN is the foundation companies use for deploying or administering value-added services including applications and data hosting network commerce, and telephony services to business customers. In private LANs, IP-based intranets have fundamentally changed the way companies conduct their business. Companies are moving their business applications to their intranets to extend over a WAN. Companies are also embracing the needs of their customers, suppliers, and partners by using extranets (an intranet that encompasses multiple businesses). With extranets, companies reduce business process costs by facilitating supply-chain automation, electronic data interchange (EDI), and other forms of network commerce. To take advantage of this business opportunity, service providers must have an IP VPN infrastructure that delivers private network services to businesses over a public infrastructure.

MPLS VPNs offer the following benefits:

- A platform for rapid deployment of additional value-added IP services, including intranets, extranets, voice, multimedia, and network commerce
- Privacy and security equal to that provided by Layer 2 VPNs by limiting the distribution of a VPN's routes to only those routers that are members of the VPN seamless integration with customer intranets
- Increased scalability over current VPN implementations, with thousands of sites per VPN and hundreds of thousands of VPNs per service provider IP class of service (CoS), with support for multiple classes of service and priorities within VPNs, as well as between VPNs
- Management of VPN membership and provisioning of new VPNs for rapid deployment
- Scalable any-to-any connectivity for extended intranets and extranets that encompass multiple businesses

Feature History for MPLS Virtual Private Networks

Feature History	
Release	**Modification**
12.0(5)T	This feature was introduced.
12.0(21)ST	This feature was implemented on the Cisco 10720 Internet router and integrated into Cisco IOS Release 12.0(21)ST.

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

12.0(22)S	This feature was implemented on the Cisco 12000 series Internet Router on the following line cards: the 6E3-SMB and 12E3-SMB line cards, the 6-port channelized T3 (6CT3-SMB) line card, the OC-192c/STM-64c Packet-over-SONET (POS) line card, and the Quad OC-48c STM-16c POS line card and integrated into Cisco IOS Release 12.0(22)S.
12.0(23)S	This feature was integrated into Cisco IOS Release 12.0(23)S. The **ip route static inter-vrf** command was introduced.
12.2(13)T	This feature was implemented on the Cisco 7200 and Cisco 7500 series routers and integrated into Cisco IOS Release 12.2(13)T. Support was added for the **ip route static inter-vrf** command.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
12.0(26)S	Support was added for the **sync** keyword to the **no ip vrf** command.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Prerequisites for MPLS Virtual Private Networks

Your network must be running the following Cisco IOS services before you configure Virtual Private Network (VPN) operation:

- Multiprotocol Label Switching (MPLS) in provider backbone routers, or generic routing encapsulation (GRE) tunnel connectivity among all provider edge (PE) routers
- MPLS with VPN code in provider routers with VPN edge service routers (PE routers)
- Border Gateway Protocol (BGP) in all routers providing a VPN service
- Cisco Express Forwarding (CEF) switching in every MPLS-enabled router
- CoS feature (optional)

To effectively implement an IP VPN in your facility, ensure that your IP VPN meets the following basic requirements:

Privacy—All IP VPNs offer privacy over a shared (public) network infrastructure. Most companies use an encrypted tunnel. This is only one of several ways to provide network and data privacy.

Scalability—For proper service delivery, VPNs must scale to serve hundreds of thousands of sites and users. Besides being a managed service, VPNs are also a management tool for service providers to control access to services. One example is Closed User Groups for data and voice services.

Flexibility—IP VPNs must handle the any-to-any traffic patterns characteristic of corporate intranets and extranets, in which data no longer flows to and from a central location. VPNs must also have the inherent flexibility to add new sites quickly, connect users over different media, and meet the increasingly sophisticated transport and bandwidth requirements of new intranet applications.

Predictable Performance—Performance needs vary widely requiring different classes of service, but the common requirement is that the performance is predictable. Examples of the ranges of performance requirements include:

- Remote access for mobile users—Require widespread connectivity
- Branch offices—Require a sustained performance level because of the interactive nature of the intranet application in a branch office
- Video conferencing—Require specific performance characteristics

Information About MPLS Virtual Private Networks

To configure MPLS Virtual Private Networks (VPNs), you need to understand the following concepts:

- Virtual Private Network Operation, page 3
- MPLS Virtual Private Networks—Basis for Value-Added Services, page 5

Virtual Private Network Operation

Each Virtual Private Network (VPN) is associated with one or more VPN routing/forwarding instances (VRFs). A VRF defines the VPN membership of a customer site attached to a PE router. A VRF consists of an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol parameters that control the information that is included into the routing table.

A one-to-one relationship does not necessarily exist between customer sites and VPNs. A given site can be a member of multiple VPNs, as shown in Figure 2. However, a site can only associate with only one VRF. A customer site's VRF contains all the routes available to the site from the VPNs of which it is a member.

Packet forwarding information is stored in the IP routing table and the CEF table for each VRF. A separate set of routing and CEF tables is maintained for each VRF. These tables prevent information from being forwarded outside a VPN, and also prevent packets that are outside a VPN from being forwarded to a router within the VPN.

This section contains the following topics:

- VPN Route Target Communities, page 4
- BGP Distribution of VPN Routing Information, page 4
- MPLS Forwarding, page 4

VPN Route Target Communities

The distribution of VPN routing information is controlled through the use of VPN route target communities, implemented by Border Gateway Protocol (BGP) extended communities. Distribution of VPN routing information works as follows:

1. When a VPN route learned from a CE router is injected into BGP, a list of VPN route target extended community attributes is associated with it. Typically the list of route target community values is set from an export list of route targets associated with the VRF from which the route was learned.
2. An import list of route target extended communities is associated with each VRF. The import list defines route target extended community attributes a route must have for the route to be imported into the VRF. For example, if the import list for a particular VRF includes route target communities A, B, and C, then any VPN route that carries any of those route target extended communities—A, B, or C—is imported into the VRF.

BGP Distribution of VPN Routing Information

A service provider edge (PE) router can learn an IP prefix from a customer edge (CE) router by static configuration, through a BGP session with the CE router, or through the Routing Information Protocol (RIP) exchange with the CE router. The IP prefix is a member of the IPv4 address family. After it learns the IP prefix, the PE converts it into a VPN-IPv4 prefix by combining it with an 8-byte route distinguisher (RD). The generated prefix is a member of the VPN-IPv4 address family. It serves to uniquely identify the customer address, even if the customer site is using globally nonunique (unregistered private) IP addresses.

The route distinguisher used to generate the VPN-IPv4 prefix is specified by a configuration command associated with the VRF on the PE router.

BGP distributes reachability information for VPN-IPv4 prefixes for each VPN. BGP communication takes place at two levels: within IP domains, known as an autonomous systems (interior BGP, or IBGP) and between autonomous systems (external BGP, or EBGP). PE-PE or PE-RR (route reflector) sessions are IBGP sessions, and PE-CE sessions are EBGP sessions.

BGP propagates reachability information for VPN-IPv4 prefixes among PE routers by means of the BGP multiprotocol extensions (refer to RFC 2283, *Multiprotocol Extensions for BGP-4*) which define support for address families other than IPv4. It does this in a way that ensures that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

MPLS Forwarding

Based on routing information stored in the VRF IP routing table and VRF CEF table, packets are forwarded to their destination using MPLS.

A PE router binds a label to each customer prefix learned from a CE router and includes the label in the network reachability information for the prefix that it advertises to other PE routers. When a PE router forwards a packet received from a CE router across the provider network, it labels the packet with the label learned from the destination PE router. When the destination PE router receives the labeled packet, it pops the label and uses it to direct the packet to the correct CE router. Label forwarding across the provider backbone is based on either dynamic label switching or traffic engineered paths. A customer data packet carries two levels of labels when traversing the backbone:

1. Top label directs the packet to the correct PE router.
2. Second label indicates how that PE router should forward the packet to the CE router.

MPLS Virtual Private Networks—Basis for Value-Added Services

MPLS VPNs allow service providers to deploy scalable VPNs and build the foundation to deliver value-added services, including:

Connectionless Service—A significant technical advantage of MPLS VPNs is that they are connectionless. The Internet owes its success to its basic technology, TCP/IP. TCP/IP is built on packet-based, connectionless network paradigm. This means that no prior action is necessary to establish communication between hosts, making it easy for two parties to communicate. To establish privacy in a connectionless IP environment, current VPN solutions impose a connection-oriented, point-to-point overlay on the network. Even if it runs over a connectionless network, a VPN cannot take advantage of the ease of connectivity and multiple services available in connectionless networks. When you create a connectionless VPN, you do not need tunnels and encryption for network privacy, thus eliminating significant complexity.

Centralized Service—Building VPNs in Layer 3 allows delivery of targeted services to a group of users represented by a VPN. A VPN must give service providers more than a mechanism for privately connecting users to intranet services. It must also provide a way to flexibly deliver value-added services to targeted customers. Scalability is critical, because customers want to use services privately in their intranets and extranets. Because MPLS VPNs are seen as private intranets, you may use new IP services such as:

- Multicast
- Quality of service (QoS)
- Telephony support within a VPN
- Centralized services including content and web hosting to a VPN

You can customize several combinations of specialized services for individual customers. For example, a service that combines IP multicast with a low-latency service class enables video conferencing within an intranet.

Scalability—If you create a VPN using connection-oriented, point-to-point overlays, Frame Relay, or ATM virtual connections (VCs), the VPN's key deficiency is scalability. Specifically, connection-oriented VPNs without fully meshed connections between customer sites are not optimal. MPLS-based VPNs instead use the peer model and Layer 3 connectionless architecture to leverage a highly scalable VPN solution. The peer model requires a customer site to peer with only one PE router as opposed to all other CPE or customer edge (CE) routers that are members of the VPN. The connectionless architecture allows the creation of VPNs in Layer 3, eliminating the need for tunnels or VCs.

Other scalability issues of MPLS VPNs are due to the partitioning of VPN routes between PE routers and the further partitioning of VPN and Interior Gateway Protocol (IGP) routes between PE routers and provider (P) routers in a core network.

- PE routers must maintain VPN routes for those VPNs who are members.
- P routers do not maintain any VPN routes.

MPLS-based VPNs increase the scalability of the provider's core and ensures that no one device is a scalability bottleneck.

Security—MPLS VPNs offer the same level of security as connection-oriented VPNs. Packets from one VPN do not inadvertently go to another VPN.

Security is provided in the following areas:

- At the edge of a provider network, ensuring packets received from a customer are placed on the correct VPN.
- At the backbone, VPN traffic is kept separate. Malicious spoofing (an attempt to gain access to a PE router) is nearly impossible because the packets received from customers are IP packets. These IP packets must be received on a particular interface or subinterface to be uniquely identified with a VPN label.

Easy to Create—To take full advantage of VPNs, it must be easy for customers to create new VPNs and user communities. Because MPLS VPNs are connectionless, no specific point-to-point connection maps or topologies are required. You can add sites to intranets and extranets and form closed user groups. When you manage VPNs in this manner, it enables membership of any given site in multiple VPNs, maximizing flexibility in building intranets and extranets.

Flexible Addressing—To make a VPN service more accessible, customers of a service provider can design their own addressing plan, independent of addressing plans for other service provider customers. Many customers use private address spaces, as defined in RFC 1918, and do not want to invest the time and expense of converting to public IP addresses to enable intranet connectivity. MPLS VPNs allow customers to continue to use their present address spaces without network address translation (NAT) by providing a public and private view of the address. A NAT is required only if two VPNs with overlapping address spaces want to communicate. This enables customers to use their own unregistered private addresses, and communicate freely across a public IP network.

Integrated Class of Service (CoS) Support—CoS is an important requirement for many IP VPN customers. It provides the ability to address two fundamental VPN requirements:

- Predictable performance and policy implementation
- Support for multiple levels of service in an MPLS VPN

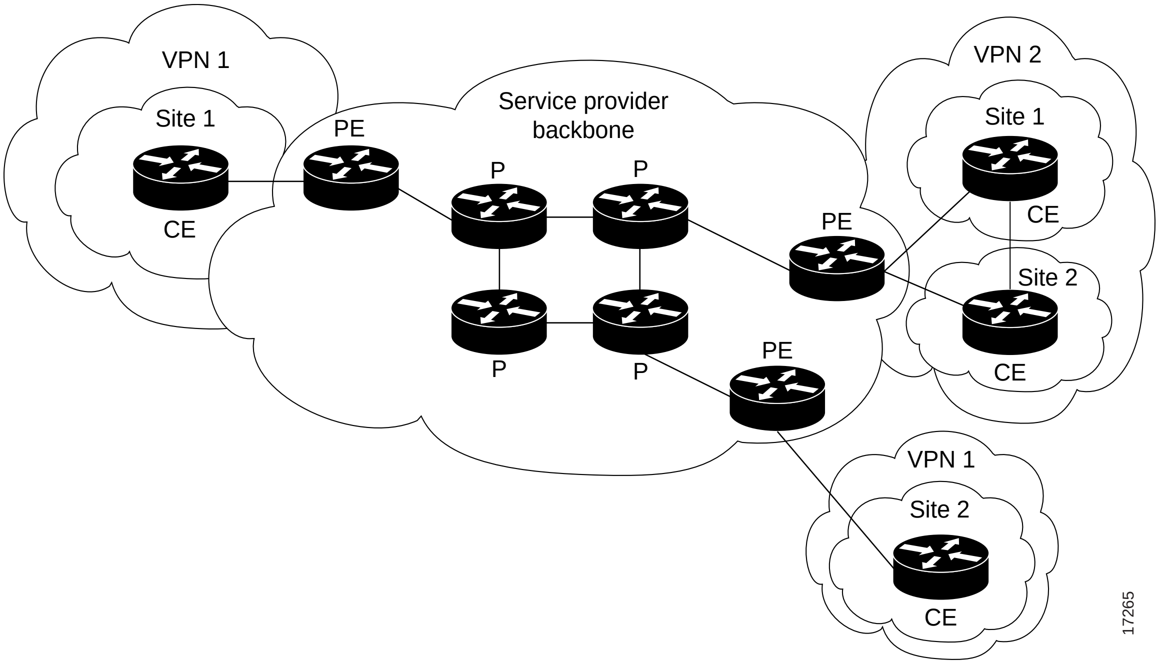
Network traffic is classified and labeled at the edge of the network before traffic is aggregated according to policies defined by subscribers and implemented by the provider and transported across the provider core. Traffic at the edge and core of the network can then be differentiated into different classes by drop probability or delay.

Straightforward Migration—For service providers to quickly deploy VPN services, use a straightforward migration path. MPLS VPNs are unique because you can build them over multiple network architectures, including IP, ATM, Frame Relay, and hybrid networks.

Migration for the end customer is simplified because there is no requirement to support MPLS on the CE router and no modifications are required to a customer's intranet.

Figure 1 shows an example of a VPN with a service provider (P) backbone network, service provider edge routers (PE), and customer edge routers (CE).

Figure 1 VPNs with a Service Provider Backbone

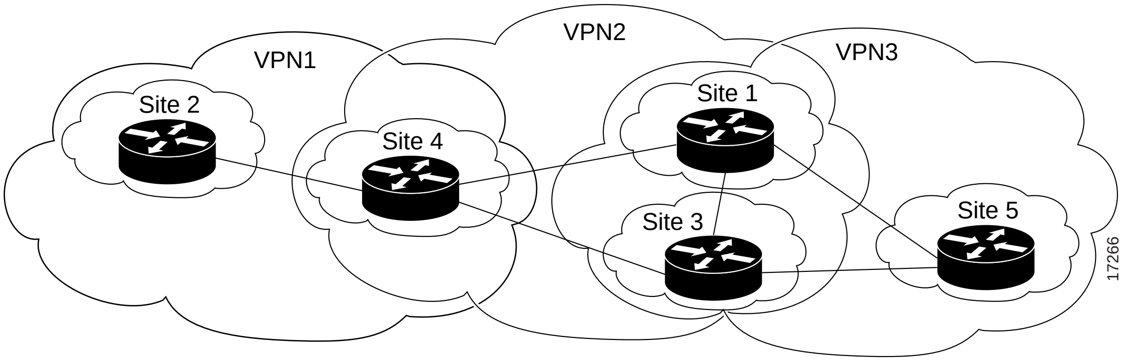


A VPN contains customer devices attached to the CE routers. These customer devices use VPNs to exchange information between devices. Only the PE routers are aware of the VPNs.

Figure 2 shows five customer sites communicating within three VPNs. The VPNs can communicate with the following sites:

- VPN 1—sites 2 and 4
- VPN 2—sites 1, 3, and 4
- VPN 3—sites 1, 3, and 5

Figure 2 Customer Sites within VPNs



How to Configure MPLS Virtual Private Networks

This section contains the following procedures to configure and verify MPLS Virtual Private Networks:

- Defining a Virtual Private Network Routing/Forwarding Instance on PE Router, page 8 (required)
- Configuring Border Gateway Protocol PE-to-PE or PE-to-CE Routing Sessions, page 10 (required)
- Configuring Routing Information Protocol PE-to-CE Routing Sessions, page 11 (required)
- Configuring Static Route PE-to-CE Routing Sessions, page 14 (required)
- Verifying Virtual Private Network Operation, page 16 (optional)
- Deleting a Virtual Private Network Routing/Forwarding Instance, page 18 (optional)

Defining a Virtual Private Network Routing/Forwarding Instance on PE Router

Perform this task to define a Virtual Private Network (VPN) routing/forwarding instance (VRF) on a provider edge (PE) router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** |**export** | **both**} *route-target-ext-community*
6. **import map** *route-map*
7. **exit**
8. **interface** *type slot/port-adapter/port* [**ethernet** | **serial**]
9. **ip vrf forwarding** *vrf-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vrf1`	Configures a VRF routing table. • The *vrf-name* argument is the name assigned to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN-IPv4 prefix.
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target import 100:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports both import and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.

Step 6	**import map** *route-map*
	Example:
	`Router(config-vrf)# import map vrf2-import`
		(Optional) Configures an import route map for a VRF.
		• The *route-map* argument specifies the route map to be used as an import route map for the VRF.

Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.
Step 8	**interface** *type slot/port-adapter/port* [**ethernet** | **serial**] **Example:** `Router(config)# interface ethernet5/0/1`	Configures an interface type and enters interface configuration mode. • The *type* argument is the type of interface to be configured. • The *slot* argument is the number of the slot being configured. • The *port-adapter* argument is the number of the port-adapter being configured. • The *port* argument is the number of the port being configured. • The **ethernet** keyword indicates an Ethernet IEEE 802.3 interface. • The **serial** keyword indicates a serial interface.

	Command or Action	Purpose
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vrf1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.

Step 10	**end**
	Example:
	`Router(config-if)# end`
		Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip vrf detail** command and make sure the MPLS VPN is up and associated with the right interfaces.

Configuring Border Gateway Protocol PE-to-PE or PE-to-CE Routing Sessions

Perform this task to configure a Border Gateway Protocol (BGP) provider edge (PE)-to-PE or a PE-to-customer edge (CE) routing session in a provider network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *peer-group-name*} **activate**
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 1`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.15.0.15 remote-as 1`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* specifies the name of a BGP peer group. • The *as-number* specifies the autonomous system to which the neighbor belongs.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router)# neighbor 10.15.0.15 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* specifies the name of a BGP peer group.

Step 6	**end**
	Example:
	`Router(config-router)# end`
		(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command is not successful, enter a **debug ip bgp *x.x.x.x* events** command, where *x.x.x.x* is the IP address of the neighbor.

Configuring Routing Information Protocol PE-to-CE Routing Sessions

Perform this task to configure a Routing Information Protocol (RIP) provider edge (PE)-to-customer edge (CE) routing session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router rip**

4. **network** *ip-address*
5. **address-family ipv4** [**multicast** | **unicast**| **vrf** *vrf-name*]
6. **exit-address-family**
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`enable`**
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	**`router rip`**
	Example:
	`Router(config)# router rip`
		Configure a RIP routing process.

Step 4	**`network`** *`ip-address`*
	Example:
	`Router(config-router)# network 10.10.0.0`
		Specifies a list of networks for the RIP routing process.
		• The *ip-address* argument specifies an IP address of the network of directly connected networks.

Step 5	**`address-family ipv4`** [**`multicast`** | **`unicast`**| **`vrf`** *`vrf-name`*] **Example:** `Router(config-router)# address-family vrf vrf1`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes. • (Optional) The **multicast** keyword specifies IPv4 multicast address prefixes. • (Optional) The **unicast** keyword specifies IPv4 unicast address prefixes. • (Optional) The **vrf** *vrf-name* keyword argument combination specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands. **Note** The default is Off for auto-summary and synchronization in the VRF address-family submode.

	Command or Action	Purpose

Step 6	**`exit-address-family`**
	Example:
	`Router(config-router-af)# exit-address-family`
		Exits address family configuration mode.

Step 7	**`end`**
	Example:
	`Router(config-router)# end`
		(Optional) Exits to privileged EXEC mode.

Configuring Static Route PE-to-CE Routing Sessions

Perform this task to configure static route provider edge (PE)-to-customer edge (CE) routing sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route vrf** *vrf-name prefix mask* [*next-hop-address*] [*interface* {*interface-number*}] [**global**] [*distance*] [**permanent**] [**tag** *tag*]
4. **address-family ipv4** [**multicast** | **unicast**| **vrf** *vrf-name*]
5. **redistribute** *protocol*
6. **exit-address-family**
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip route vrf** *vrf-name prefix mask* [*next-hop-address*] [*interface* {*interface-number*}] [**global**] [*distance*] [**permanent**] [**tag** *tag*] **Example:** `Router(config)# ip route vrf vrf1 12.0.0.0 255.0.0.0 e5/0/1 10.20.0.60`	Establishes static routes for a VRF. • The *vrf-name* argument is the name of the VRF for the static route. • The *prefix* argument specifies the IP route prefix for the destination, in dotted-decimal format. • The *mask* argument specifies the prefix mask for the destination, in dotted-decimal format. • (Optional) The *next-hop-address* argument specifies the IP address of the next hop (the forwarding router that can be used to reach that network). • (Optional) The *interface* argument specifies the type of network interface to use: ATM, Ethernet, loopback, POS (packet over SONET), or null. • (Optional) The *interface-number* argument specifies the number identifying the network interface to use. • (Optional) The **global** keyword specifies that the given next hop address is in the non-VRF routing table. • (Optional) The *distance* argument specifies an administrative distance for this route. • (Optional) The **permanent** keyword specifies that this route will not be removed, even if the interface shuts down. • (Optional) The **tag** *tag* keyword argument combination species the label (tag) value that can be used for controlling redistribution of routes through route maps.
Step 4	**address-family ipv4** [**multicast** | **unicast**| **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family vrf vrf1`	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv4 address prefixes. • (Optional) The **multicast** keyword specifies IPv4 multicast address prefixes. • (Optional) The **unicast** keyword specifies IPv4 unicast address prefixes. • (Optional) The **vrf** *vrf-name* keyword argument combination specifies the name of the VRF to associate with subsequent IPv4 address family configuration mode commands. **Note** The default is Off for auto-summary and synchronization in the VRF address-family submode.

	Command or Action	Purpose

Step 5	**`redistribute`** *`protocol`*
	Example:
	`Router(config-router-af)# redistribute static`
	Example:
	`Router(config-router-af)# redistribute connected`
		Redistributes routes from one routing domain into another routing domain.
		• The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **egp**, **igrp**, **isis**, **mobile**, **ospf**, **static** [**ip**], or **rip**.
		The **static** [**ip**] keyword is used to redistribute IP static routes. The optional **ip** keyword is used when redistributing into the IS-IS protocol.
		The **connected** keyword refers to routes that are established automatically by virtue of having enabled IP on an interface. For routing protocols such as OSPF and IS-IS, these routes will be redistributed as external to the autonomous system.

Step 6	**`exit-address-family`**
	Example:
	`Router(config-router-af)# exit-address-family`
		Exits address family configuration mode.

Step 7	**`end`**
	Example:
	`Router(config-router)# end`
		(Optional) Exits to privileged EXEC mode.

Verifying Virtual Private Network Operation

Perform this task to verify Virtual Private Network (VPN) operation.

SUMMARY STEPS

1. **enable**
2. **show ip vrf** [{**brief** | **detail** | **interfaces**}] [*vrf-name*] [*output-modifiers*]}
3. **show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]] [**summary** [*output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]]
4. **show ip protocols vrf** *vrf-name*
5. **show ip cef vrf** *vrf-name* [*ip-prefix* [*mask* [**longer-prefixes**]] [**detail**] [*output-modifiers*]] [*interface interface-number*] [**adjacency** [*interface interface-number*] [**detail**] [**discard**] [**drop**] [**glean**] [**null**] [**punt**] [*output-modifiers*]] [**detail** [*output-modifiers*]] [**non-recursive** [**detail**] [*output-modifiers*]] [**summary** [*output-modifiers*]] [**traffic** [**prefix-length**] [*output-modifiers*]] [**unresolved** [**detail**] [*output-modifiers*]]
6. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
7. **show mpls forwarding vrf** *vrf-name* [ip-prefix/length [mask]] [**detail**] [**output-modifiers**]
8. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**show ip vrf** [{**brief** | **detail** | **interfaces**}] [*vrf-name*] [*output-modifiers*]
	Example:
	`Router# show ip vrf`
	Example:
	`Router# show ip vrf interfaces`
		(Optional) Displays the set of defined VRFs and associated interfaces.
		• Use the **show ip vrf** command to display the VRFs and associated interfaces.
		• Use the **show ip vrf interfaces** command to display details about the interfaces assigned to a VRF.

Step 3	**show ip route vrf** *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]] [**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]] [**summary** [*output-modifiers*]] [**supernets-only** [*output-modifiers*]] [**traffic-engineering** [*output-modifiers*]] **Example:** `Router# show ip route vrf vpn1`	(Optional) Displays the IP routing table associated with a VRF. • Use the **show ip route vrf** command to verify that router PE1 learns routes from router CE2.
Step 4	**show ip protocols vrf** *vrf-name* **Example:** `Router# show ip protocols vrf vpn2`	(Optional) Displays the routing protocol information associated with a VRF. • Use the **show ip protocols vrf** command to check the VRF routing protocol information for PE routers.
Step 5	**show ip cef vrf** *vrf-name* [*ip-prefix* [*mask* [**longer-prefixes**]] [**detail**] [*output-modifiers*]] [*interface interface-number*] [**adjacency** [*interface interface-number*] [**detail**] [**discard**] [**drop**] [**glean**] [**null**] [**punt**] [*output-modifiers*]] [**detail** [*output-modifiers*]] [**non-recursive** [**detail**] [*output-modifiers*]] [**summary** [*output-modifiers*]] [**traffic** [**prefix-length**] [*output-modifiers*]] [**unresolved** [**detail**] [*output-modifiers*]] **Example:** `Router# show ip cef vrf vpn1`	(Optional) Displays the CEF forwarding table associated with a VRF. • Use the **show ip cef vrf** command to check the VRF CEF forwarding table on the PE routers.

	Command or Action	Purpose

Step 6	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
	Example:
	`Router# show ip bgp vpnv4 all`
	`Router# show ip bgp vpnv4 vrf vpn1 labels`
		(Optional) Displays VPN address information from the BGP table.
		• Use the **show ip bgp vpnv4 all** command to check that the BGP session is up and running between the PE and the CE routers.
		• Use the **show ip bgp vpnv4 vrf** *vrf-name* **labels** command to check that the prefixes for the provider network are in the BGP table and have the appropriate labels.

Step 7	**show mpls forwarding vrf** *vrf-name* [ip-prefix/length [mask]] [**detail**] [**output-modifiers**] **Example:** `Router# show mpls forwarding vrf vpn1 10.10.10.1 255.255.255.255 details`	(Optional) Displays label forwarding information for advertised VRF routes. • Use the **show mpls forwarding vrf** command with the **detail** keyword to check that the prefixes for the PE routers in the local customer MPLS VPN service provider are in the label forwarding information base (LFIB).
Step 8	**disable** **Example:** `Router# disable`	Exits to user EXEC mode.

Deleting a Virtual Private Network Routing/Forwarding Instance

Perform this task to delete a Virtual Private Network (VPN) routing/forwarding instance (VRF) from the router.

Virtual Private Network Routing/Forwarding Instance Deletion

When you enter the **no ip vrf** *vrf-name* command, you start the deletion of a specified VRFs. Routers delete VRFs using a background process that frees all resources associated with the VRF.

If you enter the **no ip vrf** command without the optional **sync** keyword, the command line interface (CLI) prompt returns immediately. This allows you to enter other commands while the VRF deletion process is still in progress. Any new configuration of a VRF with the same name as the VRF you deleted could get deleted and lost when the VRF resources are freed by the background process.

You can verify that the specified VRF is deleted by looking at the display of a **show ip vrf** command. If an asterisk (*) precedes the name of the VRF you deleted, then the background process has not completed (see "Deleting a Virtual Private Network Routing/Forwarding Instance Examples" section on page 24).

If you enter the **no ip vrf** command with the **sync** keyword, the router does not return the CLI prompt until the VRF deletion process is completed. This stops you from entering any commands to ensure that no new VRF configuration is lost. An informational message is displayed as the background process completes the deletion.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **no ip vrf** *vrf-name* [**sync**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# configure terminal	Enters global configuration mode.
Step 3	**no ip vrf** *vrf-name* [**sync**] **Example:** Router(config)# no ip vrf vpn1 sync	Removes a VRF routing table. • The *vrf-name* argument is the name assigned to a VRF. • The **sync** keyword blocks the CLI prompt from returning until the VRF deletion process is completed. Without the **sync** keyword, the CLI prompt returns immediately allowing you to enter new commands before the deletion process is completed.
Step 4	**end** **Example:** Router(config)# end	Returns to privileged EXEC mode.

Troubleshooting Tips

If you entered the **no ip vrf** command without the **sync** keyword, you can use the **show ip vrf** command to verify that the specified VRF is removed. An asterisk (*) before the VRF name in the command output indicates that the background process did not complete.

You can reconfigure a VRF using the name of the deleted VRF without the loss of configuration data after background processes completely remove the resources associated with the specified VRF from the router.

Configuration Examples for MPLS Virtual Private Networks

This section contains the following configuration examples for the MPLS Virtual Private Networks feature:

- Sample MPLS VPN Configuration File from a PE Router, page 20
- Defining VPN Routing Instance on PE Router Example, page 21
- Configuring BGP PE-to-PE or PE-to-CE Routing Sessions Examples, page 22
- Configuring RIP PE-to-CE Routing Sessions Example, page 22
- Configuring Static Route PE-to-CE Routing Sessions Example, page 23
- Verifying VPN Operation Examples, page 23
- Deleting a Virtual Private Network Routing/Forwarding Instance Examples, page 24

Sample MPLS VPN Configuration File from a PE Router

This section provides a sample configuration file from a PE router.

```
ip cef distributed        ! CEF switching is pre-requisite for label Switching
frame-relay switching
!
ip vrf vrf1               ! Define VPN Routing instance vrf1
rd 100:1
route-target both 100:1   ! Configure import and export route-targets for vrf1
!
ip vrf vrf2               ! Define VPN Routing instance vrf2
rd 100:2
route-target both 100:2   ! Configure import and export route-targets for vrf2
route-target import 100:1 ! Configure an additional import route-target for vrf2
import map vrf2_import    ! Configure import route-map for vrf2
!
interface lo0
ip address 10.13.0.13 255.255.255.255
!
interface atm9/0/0                    ! Backbone link to another Provider router
!
interface atm9/0/0.1 tag-switching
ip unnumbered loopback0
no ip directed-broadcast
mpls atm vpi 2-5
mpls ip

interface atm5/0
no ip address
no ip directed-broadcast
atm clock INTERNAL
no atm ilmi-keepalive

interface Ethernet1/0
ip address 3.3.3.5 255.255.0.0
no ip directed-broadcast
no ip mroute-cache
no keepalive

interface Ethernet5/0/1                  ! Set up Ethernet interface
ip vrf forwarding vrf1                   ! as VRF link to a CE router
ip address 10.20.0.13 255.255.255.0
```

```
!
interface hssi 10/1/0
hssi internal-clock
encaps fr
frame-relay intf-type dce
frame-relay lmi-type ansi
!
interface hssi 10/1/0.16 point-to-point
ip vrf forwarding vrf2
ip address 10.20.1.13 255.255.255.0
frame-relay interface-dlci 16           ! Set up Frame Relay PVC
!                                       ! subinterface as link to another
!                                       ! CE router
!
router bgp 1                            ! Configure BGP sessions
no synchronization
no bgp default ipv4-activate            ! Deactivate default IPv4 advertisements
neighbor 10.15.0.15 remote-as 1         ! Define IBGP session with another PE
neighbor 10.15.0.15 update-source lo0
!
address-family vpnv4 unicast            ! Activate PE exchange of VPNv4 NLRI
neighbor 10.15.0.15 activate
exit-address-family
!
address-family ipv4 unicast vrf vrf1    ! Define BGP PE-CE session for vrf1
redistribute static
redistribute connected
neighbor 10.20.0.60 remote-as 65535
neighbor 10.20.0.60 activate
no auto-summary
exit-address-family
!
address-family ipv4 unicast vrf vrf2    ! Define BGP PE-CE session for vrf2
redistribute static
redistribute connected
neighbor 10.20.1.11 remote-as 65535
neighbor 10.20.1.11 update-source h10/1/0.16
neighbor 10.20.1.11 activate
no auto-summary
exit-address-family
!
! Define a VRF static route
ip route vrf vrf1 12.0.0.0 255.0.0.0 e5/0/1 10.20.0.60
!
route-map vrf2_import permit 10 ! Define import route-map for vrf2.
...
```

Defining VPN Routing Instance on PE Router Example

This example shows the configuration of VPN routing instances on a PE router:

```
ip cef distributed        ! CEF switching is pre-requisite for label Switching
frame-relay switching
!
ip vrf vrf1               ! Define VPN Routing instance vrf1
rd 100:1
route-target both 100:1   ! Configure import and export route-targets for vrf1
!
ip vrf vrf2               ! Define VPN Routing instance vrf2
rd 100:2
```

```
route-target both 100:2   ! Configure import and export route-targets for vrf2
route-target import 100:1 ! Configure an additional import route-target for vrf2
import map vrf2_import    ! Configure import route-map for vrf2
!
```

Configuring BGP PE-to-PE or PE-to-CE Routing Sessions Examples

This example shows the configuration of a BGP PE-to-PE routing session:

```
router bgp 1                            ! Configure BGP sessions
no synchronization
no bgp default ipv4-activate            ! Deactivate default IPv4 advertisements
neighbor 10.15.0.15 remote-as 1         ! Define IBGP session with another PE
neighbor 10.15.0.15 update-source lo0
!
address-family vpnv4 unicast            ! Activate PE exchange of VPNv4 NLRI
neighbor 10.15.0.15 activate
exit-address-family
!
```

This example shows the configuration of a BGP PE-to-CE session for vrf1:

```
address-family ipv4 unicast vrf vrf1     ! Define BGP PE-CE session for vrf1
redistribute static
redistribute connected
neighbor 10.20.0.60 remote-as 65535
neighbor 10.20.0.60 activate
no auto-summary
exit-address-family
!
```

This example shows the configuration of a BGP PE-to-CE session for vrf2:

```
address-family ipv4 unicast vrf vrf2     ! Define BGP PE-CE session for vrf2
redistribute static
redistribute connected
neighbor 10.20.1.11 remote-as 65535
neighbor 10.20.1.11 update-source h10/1/0.16
neighbor 10.20.1.11 activate
no auto-summary
exit-address-family
!
```

Configuring RIP PE-to-CE Routing Sessions Example

This example shows the configuration of a RIP PE-to-CE routing session for vrf1:

```
router rip
  version 2
  !
  address-family ipv4 vrf vrf1
  version 2
  redistribute bgp 1 metric 0
  network 10.0.13.0
  no auto-summary
  exit-address-family
```

Configuring Static Route PE-to-CE Routing Sessions Example

This example shows the configuration of a static routing session between a PE and CE router:

```
ip route vrf vrf1 12.0.0.0 255.0.0.0 e5/0/1 10.20.0.60
!
route-map vrf2_import permit 10 ! Define import route-map for vrf2.
...
```

Verifying VPN Operation Examples

The output of the **show ip vrf** command shows the VRFs currently configured:

```
Router# show ip vrf

  Name                Default RD          Interfaces
  vrf1                100:1               Ethernet1/3
  vrf2                100:2               Ethernet0/3
```

The output of the **show ip vrf interfaces** command shows the interfaces bound to a particular VRF:

```
Router# show ip vrf interfaces

Interface        IP-Address       VRF                        Protocol
Ethernet2        130.22.0.33      blue_vrf                   up
Ethernet4        130.77.0.33      hub                        up
router#
```

The output of the **show ip route vrf vpn1** command shows the IP routing table associated with the VRF called vpn1:

```
Router# show ip route vrf vpn1

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

B    51.0.0.0/8 [200/0] via 13.13.13.13, 00:24:19
C    50.0.0.0/8 is directly connected, Ethernet1/3
B    11.0.0.0/8 [20/0] via 50.0.0.1, 02:10:22
B    12.0.0.0/8 [200/0] via 13.13.13.13, 00:24:20
```

The output of the **show ip route vrf vpn2** command displays information about a VRF called vpn2:

```
Router# show ip protocols vrf vpn2

Routing Protocol is "bgp 100"
  Sending updates every 60 seconds, next due in 0 sec
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Redistributing:connected, static
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
    13.13.13.13          200      02:20:54
    18.18.18.18          200      03:26:15
  Distance:external 20 internal 200 local 200
```

The output of the **show ip cef vrf vpn1** command shows the forwarding table associated with the VRF called vpn1:

```
Router# show ip cef vrf vpn1

Prefix              Next Hop            Interface
0.0.0.0/32          receive
11.0.0.0/8          50.0.0.1            Ethernet1/3
12.0.0.0/8          52.0.0.2            POS6/0
50.0.0.0/8          attached            Ethernet1/3
50.0.0.0/32         receive
50.0.0.1/32         50.0.0.1            Ethernet1/3
50.0.0.2/32         receive
50.255.255.255/32   receive
51.0.0.0/8          52.0.0.2            POS6/0
224.0.0.0/24        receive
255.255.255.255/32  receive
```

The output of the **show ip bgp vpnv4 all** command shows all VPNv4 information in a BGP routing table:

```
Router# show ip bgp vpnv4 all

BGP table version is 18, local router ID is 14.14.14.14
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
Route Distinguisher: 100:1 vrf1
*> 11.0.0.0         50.0.0.1           0        0                101 i
*>i12.0.0.0         13.13.13.13        0        100      0       102 i
*> 50.0.0.0         50.0.0.1           0        0                101 i
*>i51.0.0.0         13.13.13.13        0        100      0       102 i
```

Deleting a Virtual Private Network Routing/Forwarding Instance Examples

The following example shows the removal of a VRF with the **sync** keyword that blocks the return of the command prompt until the process is completed:

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# no ip vrf vpn5 ?
  sync  Return after completing VRF delete
  <cr>

Router(config)# no ip vrf vpn5 sync

% IP addresses from all interfaces in VRF vpn5 have been removed

Router(config)# end
```

The following example shows the VRF configuration on the router before entering the **no ip vrf vpn5 sync** command:

```
Router# show ip vrf

  Name                             Default RD          Interfaces
  forw                             <not set>
  mgmt                             200:1
  vpn5                             500:1               Ethernet1/4
  vpn6                             600:1               Ethernet1/6
```

The following example shows the VRF configuration on the router after entering the **no ip vrf vpn5 sync** command:

```
Router# show ip vrf

  Name                             Default RD          Interfaces
  forw                             <not set>
  mgmt                             200:1
  vpn6                             600:1               Ethernet1/6
```

The following example shows the removal of a VRF without a prompt blocking option:

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# no ip vrf vpn6

% IP addresses from all interfaces in VRF vpn6 have been removed

Router(config)# end

00:03:34: %OSPF-5-ADJCHG: Process 66, Nbr 33.33.33.33 on Ethernet1/6 from FULL
to DOWN, Neighbor Down: Interface down or detached
```

The following example shows the VRF is in the process of being deleted:

```
Router# show ip vrf
  Name                             Default RD          Interfaces
  forw                             <not set>
  mgmt                             200:1
* vpn6                             600:1

* Being deleted

Router#
00:03:35: %SYS-5-CONFIG_I: Configured from console by console
```

The following example shows reconfiguring a VRF using the same name (vpn6) as the VRF just deleted:

```
Router# ip vrf vpn6
Router(config-vrf)# rd 600:1
Router(config-vrf)# route-target both 600:1
Router(config-vrf)# route-target import 600:2
```

The following example shows configuration lost as a result of entering commands before the deletion process is completed:

```
Router# show ip vrf
  Name                             Default RD          Interfaces
  forw                             <not set>
  mgmt                             200:1
```

Additional References

The following sections provide references related to MPLS VPNs:

Related Documents

Related Topic	Document Title
Enhanced MPLS VPN traffic management configuration tasks	*MPLS Virtual Private Network Enhancements*
MPLS CoS definition and configuration tasks	*MPLS Class of Service (CoS)*
MPLS CoS enhancement configuration tasks	*MPLS Class of Service Enhancements*
MPLS forwarding configuration tasks	*Multiprotocol Label Switching (MPLS) on Cisco Routers*
MPLS Label Distribution Protocol (LDP) configuration tasks	*MPLS Label Distribution Protocol (LDP)*
BGP configuration tasks	“Configuring BGP chapter” in the *Cisco IOS IP Configuration Guide*, Release 12.2
OSPF configuration tasks	“Configuring OSFP” chapter in the *Cisco IOS IP Configuration Guide, Release 12.2, IP Routing Protocols*
IS-IS configuration tasks	“Configuring Integrated IS-IS chapter” in the *Cisco IOS IP Configuration Guide, Release 12.2, IP Routing Protocols*

Standards

Standards	Title
No new standards or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new MIBs or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1163	*A Border Gateway Protocol*
RFC 1164	*Application of the Border Gateway Protocol in the Internet*
RFC 2283	*Multiprotocol Extensions for BGP-4*
RFC 2547	*BGP/MPLS VPNs*

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents new and modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command references.

- **address-family**
- **clear ip route vrf**
- **debug ip bgp**
- **exit-address-family**
- **import map**
- **ip route static inter-vrf**
- **ip route vrf**
- **ip vrf**
- **ip vrf forwarding**
- **neighbor activate**
- **rd**
- **route-target**
- **show ip bgp vpnv4**
- **show ip cef vrf**
- **show ip protocols vrf**
- **show ip route vrf**
- **show ip vrf**
- **show mpls forwarding vrf**

address-family

To enter the address family submode for configuring routing protocols, such as Border Gateway Protocol (BGP), Routing Information Protocol (RIP) and static routing, use the **address-family** command in router configuration mode. To disable the address family submode for configuring routing protocols, use the **no** form of this command.

VPN-IPv4 unicast

address-family vpnv4 [**unicast**]

no address-family vpnv4 [**unicast**]

IPv4 unicast

address-family ipv4 [**unicast**]

no address-family ipv4 [**unicast**]

IPv4 unicast with CE router

address-family ipv4 [**unicast**] **vrf** *vrf-name*

no address-family ipv4 [**unicast**] **vrf** *vrf-name*

Syntax Description

ipv4	Configures sessions that carry standard IPv4 address prefixes.
vpnv4	Configures sessions that carry customer VPN-IPv4 prefixes, each of which has been made globally unique by adding an 8-byte route distinguisher.
unicast	(Optional) Specifies unicast prefixes.
vrf *vrf-name*	Specifies the name of a VPN routing/forwarding instance (VRF) to associate with submode commands.

Defaults

Routing information for address family IPv4 is advertised by default when you configure a BGP session using the **neighbor remote-as** command unless you execute the **no bgp default ipv4-activate** command.

Command Modes

Router configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.

Release	Modification
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Using the **address-family** command puts you in address family configuration mode. Within this mode, you can configure address-family specific parameters for routing protocols, such as BGP, that can accommodate multiple Layer 3 address families.

To leave address family configuration submode and return to router configuration mode, type **exit-address-family**, or simply **exit**.

Examples

The following example shows how to put the router into address family configuration submode for the VPNv4 address family. Within the submode, you can configure advertisement of Network Layer Reachability Information (NLRI) for the VPNv4 address family using **neighbor activate** and other related commands:

```
Router(config)# router bgp 100

Router(config-router)# address-family vpnv4

Router(config-router-af)#
```

The following example shows how to put the router into address family configuration submode for the IPv4 address family. Use this form of the command, which specifies a VRF, only to configure routing exchanges between provider edge (PE) and customer edge (CE) devices. This address-family command causes subsequent commands entered in the submode to be executed in the context of VRF vrf2.

```
Router(config)# router bgp 100

Router(config-router)# address-family ipv4 unicast vrf vrf2

Router(config-router-af)#
```

Within the submode, you can use **neighbor activate** and other related commands to accomplish the following:

- Configure advertisement of IPv4 NLRI between the PE and CE routers.
- Configure translation of the IPv4 NLRI (that is, translate IPv4 into VPNv4 for NLRI received from the CE, and translate VPNv4 into IPv4 for NLRI to be sent from the PE to the CE).
- Enter the routing parameters that apply to this VRF.

Related Commands

Command	Description
exit-address-family	Exits from the address family submode.
neighbor activate	Enables the exchange of information with a BGP neighboring router.

clear ip route vrf

To remove routes from the Virtual Private Network (VPN) routing/forwarding instance (VRF) routing table, use the **clear ip route vrf** command in privileged EXEC mode.

clear ip route vrf *vrf-name* {***** | *network* [*mask*]}

Syntax Description

vrf-name	Name of the VRF for the static route.
*****	Deletes all routes for a given VRF.
network	Destination to be removed, in dotted-decimal format.
mask	(Optional) Mask for the specified network destination, in dotted-decimal format.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modifications
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use this command to clear routes from the routing table. Use the asterisk (*) to delete all routes from the forwarding table for a specified VRF, or enter the address and mask of a particular network to delete the route to that network.

Examples

The following command shows how to remove the route to the network 10.13.0.0 in the vpn1 routing table:

```
Router# clear ip route vrf vpn1 10.13.0.0
```

Related Commands

Command	Description
show ip route vrf	Displays the IP routing table associated with a VRF.

debug ip bgp

To display information related to processing Border Gateway Protocol (BGP) routing, use the **debug ip bgp** command in privileged EXEC mode. To disable the display of BGP information, use the **no** form of this command.

debug ip bgp [*A.B.C.D.* | **dampening** | **events** | **in** | **keepalives** | **out** | **updates** | **vpnv4**]

no debug ip bgp [*A.B.C.D.* | **dampening** | **events** | **in** | **keepalives** | **out** | **updates** | **vpnv4**]

Syntax Description

A.B.C.D.	(Optional) Displays the BGP neighbor IP address.
dampening	(Optional) Displays BGP dampening.
events	(Optional) Displays BGP events.
in	(Optional) BGP inbound information.
keepalives	(Optional) Displays BGP keepalives.
out	(Optional) Displays BGP outbound information.
updates	(Optional) Displays BGP updates.
vpnv4	(Optional) Displays VPNv4 Network Layer Reachability Information (NLRI) information.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Examples

The following example displays the output from this command:

```
Router# debug ip bgp vpnv4

03:47:14:vpn:bgp_vpnv4_bnetinit:100:2:58.0.0.0/8
03:47:14:vpn:bnettable add:100:2:58.0.0.0 / 8
03:47:14:vpn:bestpath_hook route_tag_change for vpn2:58.0.0.0/255.0.0.0(ok)
03:47:14:vpn:bgp_vpnv4_bnetinit:100:2:57.0.0.0/8
03:47:14:vpn:bnettable add:100:2:57.0.0.0 / 8
03:47:14:vpn:bestpath_hook route_tag_change for vpn2:57.0.0.0/255.0.0.0(ok)
03:47:14:vpn:bgp_vpnv4_bnetinit:100:2:14.0.0.0/8
03:47:14:vpn:bnettable add:100:2:14.0.0.0 / 8
03:47:14:vpn:bestpath_hook route_tag_chacle ip bgp *nge for vpn2:14.0.0.0/255.0.0.0(ok)
```

exit-address-family

To exit from the address family submode, use the **exit-address-family** command in address family submode.

exit-address-family

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

Address family submode

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

This command can be abbreviated to **exit**.

Examples

The following example shows how to exit the address family configuration submode:

```
Router(config-router-af)# exit-address-family
```

Related Commands

Command	Description
address-family	Enters the address family submode used to configure routing protocols.

import map

To configure an import route map for a Virtual Private Network (VPN) routing/forwarding instance (VRF), use the **import map** command in VRF submode.

import map *route-map*

Syntax Description

route-map	Specifies the route map to be used as an import route map for the VRF.

Defaults

A VRF has no import route map unless one is configured using the **import map** command.

Command Modes

VRF submode

Command History

Command	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use an **import map** command when an application requires finer control over the routes imported into a VRF than provided by the import and export extended communities configured for the importing and exporting VRF.

The **import map** command associates a route map with the specified VRF. You can filter routes that are eligible for import into a VRF, based on the route target extended community attributes of the route, through the use of a route map. The route map might deny access to selected routes from a community that is on the import list.

Examples

The following example shows how to configure an import route map for a VRF:

```
Router(config)# ip vrf vrf_blue

Router(config-vrf)# import map blue_import_map
```

Related Commands

Command	Description
ip vrf	Enters VRF configuration mode.
route-target	Configures import and export extended community attributes for the VRF.
show ip vrf	Displays information about a VRF or all VRFs.

ip route static inter-vrf

To allow static routes to point to Virtual Private Network (VPN) routing/forwarding instance (VRF) interfaces in VRFs other than those to which the static route belongs, use the **ip route static inter-vrf** command in global configuration mode. To prevent static routes from pointing to VRF interfaces in VRFs to which they do not belong, use the **no** form of this command.

ip route static inter-vrf

no ip route static inter-vrf

Syntax Description

This command has no arguments or keywords.

Defaults

By default, static routes are allowed to point to VRF interfaces in any VRF.

Command Modes

Global configuration

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

The **ip route static inter-vrf** command is turned on by default. The **no ip route static inter-vrf** command causes the respective routing table (global or VRF) to reject the installation of static routes if the outgoing interface belongs to a different VRF than the static route being configured. This prevents security problems that can occur when static routes that point to a VRF interface in a different VRF are misconfigured. You are notified when a static route is rejected, then you can reconfigure it.

For example, a static route is defined on a provider edge (PE) router to forward Internet traffic to a customer on the interface pos1/0, as follows:

```
Router(config)# ip route 10.1.1.1 255.255.255.255 pos1/0
```

Mistakenly, the same route is configured with the next-hop as the VRF interface pos10/0:

```
Router(config)# ip route 10.1.1.1 255.255.255.255 pos10/0
```

By default, Cisco IOS accepts the command and starts forwarding the traffic to both pos1/0 (Internet) and pos10/0 (VPN) interfaces.

If the static route is already configured that points to a VRF other than the one to which the route belongs when you issue the **no ip route static inter-vrf** command, the offending route is uninstalled from the routing table and a message similar to the following is sent to the console:

```
01:00:06: %IPRT-3-STATICROUTESACROSSVRF: Un-installing static route x.x.x.x/32 from global
routing table with outgoing interface intx/x
```

If you enter the **no ip route static inter-vrf** command before a static route is configured that points to a VRF interface in a different VRF, the static route is not installed in the routing table and a message is sent to the console.

In the following example, configuring the **no ip route static inter-vrf** command prevents traffic from following an unwanted path. A VRF static route points to a global interface or any other VRF interface as shown in the following **ip route vrf** commands:

- Interface ser1/0.0 is a global interface:

```
Router(config)# no ip route static inter-vrf

Router(config)# ip route vrf vpn1 10.10.1.1 255.255.255.255 ser1/0.0
```

- Interface ser1/0.1 is in vpn2:

```
Router(config)# no ip route static inter-vrf

Router(config)# ip route vrf vpn1 10.10.1.1 255.255.255.255 ser1/0.1
```

With the **no ip route static inter-vrf** command configured, these static routes are not installed into the vpn1 routing table because the static routes point to an interface that is not in the same VRF.

If you require a VRF static route to point to a global interface, you can use the **global** keyword with the **ip route vrf** command:

```
Router(config)# ip route vrf vpn1 10.12.1.1 255.255.255.255 ser1/0.0 7.0.0.1 global
```

The **global** keyword allows the VRF static route to point to a global interface even when the **no ip route static inter-vrf** command is configured.

Examples

The following example shows how to prevent static routes that point to VRF interfaces in a different VRF:

```
Router(config)# no ip route static inter-vrf
```

Related Commands

Command	Description
ip route vrf	Establishes static routes for a VRF.

ip route vrf

To establish static routes for a Virtual Private Network (VPN) routing/forwarding instance (VRF), use the **ip route vrf** command in global configuration mode. To disable static routes, use the **no** form of this command.

ip route vrf *vrf-name prefix mask* [*next-hop-address*] [*interface* {*interface-number*}] [**global**] [*distance*] [**permanent**] [**tag** *tag*]

no ip route vrf *vrf-name prefix mask* [*next-hop-address*] [*interface* {*interface-number*}] [**global**] [*distance*] [**permanent**] [**tag** *tag*]

Syntax Description

vrf-name	Name of the VRF for the static route.
prefix	IP route prefix for the destination, in dotted-decimal format.
mask	Prefix mask for the destination, in dotted-decimal format.
next-hop-address	(Optional) IP address of the next hop (the forwarding router that can be used to reach that network).
interface	(Optional) Type of network interface to use: ATM, Ethernet, loopback, POS (packet over SONET), or null.
interface-number	Number identifying the network interface to use.
global	Specifies that the given next hop address is in the non-VRF routing table.
distance	(Optional) An administrative distance for this route.
permanent	(Optional) Specifies that this route will not be removed, even if the interface shuts down.
tag *tag*	(Optional) Label value that can be used for controlling redistribution of routes through route maps.

Defaults

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modifications
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use a static route when the Cisco IOS software cannot dynamically build a route to the destination.

If you specify an administrative distance when you set up a route, you are flagging a static route that can be overridden by dynamic information. For example, Interior Gateway Routing Protocol (IGRP)-derived routes have a default administrative distance of 100. To set a static route to be overridden by an IGRP dynamic route, specify an administrative distance greater than 100. Static routes each have a default administrative distance of 1.

Static routes that point to an interface are advertised through Routing Information Protocol (RIP), IGRP, and other dynamic routing protocols, regardless of whether the routes are redistributed into those routing protocols. That is, static routes configured by specifying an interface lose their static nature when installed into the routing table.

However, if you define a static route to an interface not defined in a network command, no dynamic routing protocols advertise the route unless a redistribute static command is specified for these protocols.

Examples

The following command shows how to reroute packets addressed to network 137.23.0.0 in VRF vpn3 to router 131.108.6.6:

```
Router(config)# ip route vrf vpn3 137.23.0.0 255.255.0.0 131.108.6.6
```

Related Commands

Command	Description
show ip route vrf	Displays the IP routing table associated with a VRF.

ip vrf

To configure a Virtual Private Network (VPN) routing/forwarding instance (VRF) routing table, use the **ip vrf** command in global configuration mode. To remove a VRF routing table, use the **no** form of this command.

ip vrf *vrf-name*

no ip vrf *vrf-name* [**sync**]

Syntax Description

vrf-name	Name assigned to a VRF.
sync	(Optional) Used only with the **no** form of the command to block the command line interface (CLI) prompt from returning until the VRF deletion process is completed. Without the **sync** keyword, the CLI prompt returns immediately allowing you to enter new commands before the deletion process is completed.

Defaults

No VRFs are defined. No import or export lists are associated with a VRF. No route maps are associated with a VRF.

Command Modes

Global configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.
12.0(26)S	The optional **sync** keyword was added for use with the **no ip vrf** command.

Usage Guidelines

The **ip vrf** *vrf-name* command creates a VRF routing table and a Cisco Express Forwarding (CEF) table, both named *vrf-name*. Associated with these tables is the default route distinguisher value *route-distinguisher.*

Use the **sync** keyword to prevent the loss of VRF configuration when you delete a specified VRF with the **no ip vrf** command and reconfigure a new VRF within a few minutes using the same name as the just deleted VRF. The **sync** keyword blocks the command prompt so that you cannot enter any new configuration commands until the router’s background process completely frees the resources associated with the specified VRF.

Examples

The following example shows how to configure a VRF routing table named vpn1:

```
Router# configure terminal
Router(config)# ip vrf vpn1
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target both 100:2
Router(config-vrf)# route-target import 100:1
```

The following example shows how to prevent the loss of VRF configuration when reconfiguring a VRF with the same name as a recently deleted VRF:

```
Router# configure terminal
Router(config)# no ip vrf vpn1 sync

% IP addresses from all interfaces in VRF vpn1 have been removed

Router(config)# end

Router# configure terminal
Router(config)# ip vrf vpn1
Router(config-vrf)#
```

Note Use the **show ip vrf** command to verify that the specified VRF is deleted.

Related Commands

Command	Description
ip vrf forwarding	Associates a VRF with an interface or subinterface.

ip vrf forwarding

To associate a Virtual Private Network (VPN) routing/forwarding instance (VRF) with an interface or subinterface, use the **ip vrf forwarding** command in interface configuration mode. To disassociate a VRF, use the **no** form of this command.

ip vrf forwarding *vrf-name*

no ip vrf forwarding *vrf-name*

Syntax Description

vrf-name	Name assigned to a VRF.

Defaults

The default for an interface is the global routing table.

Command Modes

Interface configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use this command to associate an interface with a VRF. Executing this command on an interface removes the IP address. The IP address should be reconfigured.

Examples

The following example shows how to link a VRF to ATM interface 0/0:

```
Router(config)# interface atm0/0

Router(config-if)# ip vrf forwarding vpn1
```

Related Commands

Command	Description
ip vrf	Defines a VRF.
ip route vrf	Establishes static routes for a VRF.

neighbor activate

To enable the exchange of information with a Border Gateway Protocol (BGP) neighboring router, use the **neighbor activate** command in router configuration mode. To disable the exchange of an address with a neighboring router, use the **no** form of this command.

neighbor {*ip-address* | *peer-group-name*} **activate**

no neighbor {*ip-address* | *peer-group-name*} **activate**

Syntax Description

ip-address	IP address of the neighboring router.
peer-group-name	Name of BGP peer group.

Defaults

The exchange of addresses with neighbors is enabled by default for the Virtual Private Network (VPN) IPv4 address family. You can disable IPv4 address exchange using the general command **no default bgp ipv4 activate**, or you can disable it for a particular neighbor by using the **no** form of this command.

For all other address families, address exchange is disabled by default. You can explicitly activate the default command by using the appropriate address family configuration submode.

Command Modes

Router configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use this command to enable or disable the exchange of addresses with a neighboring router.

Examples

The following example shows how to activate the exchange of the customer IP address 10.15.0.15 to a neighboring router:

```
Router(config)# router bgp 100

Router(config-router)# neighbor 10.15.0.15 remote-as 100

Router(config-router)# neighbor 10.15.0.15 update-source loopback0
Router(config-router)# address-family vpnv4 unicast

Router(config-router-af)# neighbor 10.15.0.15 activate

Router(config-router-af)# exit-address-family
```

Related Commands

Command	Description
address-family	Enters the address family submode.
exit-address-family	Exits the address family submode.

rd

To create routing and forwarding tables for a Virtual Private Network (VPN) routing/forwarding instance (VRF), use the **rd** command in VRF configuration submode.

rd *route-distinguisher*

Syntax Description

route-distinguisher	Adds an 8-byte value to an IPv4 prefix to create a VPN-IPv4 prefix.

Defaults

There is no default. A route distinguisher (RD) must be configured for a VRF to be functional.

Command Modes

VRF configuration submode

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

An RD creates routing and forwarding tables and specifies the default route distinguisher for a VPN. The RD is added to the beginning of the customer's IPv4 prefixes to change them into globally unique VPN-IPv4 prefixes.

An RD is either

- ASN-related—Composed of an autonomous system number and an arbitrary number.
- IP-address-related—Composed of an IP address and an arbitrary number.

You can enter an RD in either of these formats:

16-bit AS number: *your 32-bit number*
For example, 101:3

32-bit IP address: *your 16-bit number*
For example, 192.168.122.15:1

Examples

The following example shows how to configure a default RD for two VRFs. The example shows the use of both AS-related and IP address-related RDs:

```
Router(config)# ip vrf vrf_blue

Router(config-vrf)# rd 100:3

Router(config-vrf)# ip vrf vrf_red

Router(config-vrf)# rd 173.13.0.12:200
```

Related Commands

Command	Description
rd	Enters VRF configuration mode.
show ip vrf	Displays information about a VRF.

route-target

To create a route-target extended community for a Virtual Private Network (VPN) routing/forwarding instance (VRF), use the **route-target** command in VRF configuration submode. To disable the configuration of a route-target community option, use the **no** form of this command.

route-target {**import** | **export** | **both**} *route-target-ext-community*

no route-target {**import** | **export** | **both**} *route-target-ext-community*

Syntax Description

import	Imports routing information from the target VPN extended community.
export	Exports routing information to the target VPN extended community.
both	Imports both import and export routing information to the target VPN extended community.
route-target-ext-community	Adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.

Defaults

A VRF has no route-target extended community attributes associated with it until the attributes are specified by the **route-target** command.

Command Modes

VRF configuration submode

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

The **route-target** command creates lists of import and export route target extended communities for the specified VRF. Execute the command one time for each target community. Learned routes that carry a specific route target extended community are imported into all VRFs configured with that extended community as an import route target. Routes learned from a VRF site (for example, by Border Gateway Protocol (BGP), Routing Information Protocol (RIP), or static route configuration) contain export route targets for extended communities configured for the VRF added as route attributes to control the VRFs into which the route is imported.

The route-target specifies a target VPN extended community. Like a route-distinguisher, an extended community is composed of either an autonomous system number and an arbitrary number, or an IP address and an arbitrary number. You can enter the numbers in either of these formats:

- *16-bit AS number*: *your 32-bit number*
  For example, 101:3
- *32-bit IP address*: *your 16-bit number*
  For example, 192.168.122.15:1

Examples

The following example shows how to configure route-target extended community attributes for a VRF. The result of the command sequence is that VRF *vrf_blue* has two export extended communities (1000:1 and 1000:2) and two import extended communities (1000:1 and 173.27.0.130:200).

```
Router(config)# ip vrf vrf_blue

Router(config-vrf)# route-target both 1000:1

Router(config-vrf)# route-target export 1000:2

Router(config-vrf)# route-target import 173.27.0.130:200
```

Related Commands

Command	Description
import map	Configures an import route map for the VRF.
ip vrf	Enters VRF configuration mode.

show ip bgp vpnv4

To display Virtual Private Network (VPN) address information from the Border Gateway Protocol (BGP) table, use the **show ip bgp vpnv4** command in privileged EXEC mode.

show ip bgp vpnv4 {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix/length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] **[dampened-paths**] [**filter-list**] **[flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**labels**]

Syntax Description

all	Displays the complete VPNv4 database.
rd *route-distinguisher*	Displays NLRIs that have a matching route distinguisher.
vrf *vrf-name*	Displays NLRIs associated with the named VRF.
ip-prefix/length	(Optional) IP prefix address (in dotted decimal format) and length of mask (0 to 32).
longer-prefixes	(Optional) Displays the entry, if any, that exactly matches the specified prefix parameter, as well as all entries that match the prefix in a “longest-match” sense. That is, prefixes for which the specified prefix is an initial substring.
output-modifiers	(Optional) For a list of associated keywords and arguments, use context-sensitive help.
network-address	(Optional) IP address of a network in the BGP routing table.
mask	(Optional) Mask of the network address, in dotted decimal format.
cidr-only	(Optional) Displays only routes that have nonnatural net masks.
community	(Optional) Displays routes matching this community.
community-list	(Optional) Displays routes matching this community list.
dampened-paths	(Optional) Displays paths suppressed due to dampening (BGP route from peer is up and down).
filter-list	(Optional) Displays routes conforming to the filter list.
flap-statistics	(Optional) Displays flap statistics of routes.
inconsistent-as	(Optional) Displays only routes that have inconsistent autonomous systems of origin.
neighbors	(Optional) Displays details about TCP and BGP neighbor connections.
paths	(Optional) Displays path information.
line	(Optional) A regular expression to match the BGP AS paths.
peer-group	(Optional) Displays information about peer groups.
quote-regexp	(Optional) Displays routes matching the AS path “regular expression.”
regexp	(Optional) Displays routes matching the AS path regular expression.
summary	(Optional) Displays BGP neighbor status.
labels	(Optional) Displays incoming and outgoing BGP labels for each NLRI.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use this command to display VPNv4 information from the BGP database. The **show ip bgp vpnv4 all** command displays all available VPNv4 information. The **show ip bgp vpnv4 summary** command displays BGP neighbor status.

Examples

The following example shows output for all available VPNv4 information in a BGP routing table:

```
Router# show ip bgp vpnv4 all

BGP table version is 18, local router ID is 14.14.14.14
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
Route Distinguisher: 100:1 vrf1
*> 11.0.0.0         50.0.0.1           0        0                101 i
*>i12.0.0.0         13.13.13.13        0        100      0       102 i
*> 50.0.0.0         50.0.0.1           0        0                101 i
*>i51.0.0.0         13.13.13.13        0        100      0       102 i
```

Table 1 describes the fields shown in the example.

Table 1 show ip bgp vpnv4 Field Descriptions

Field	Description
Network	Displays the network address from the BGP table.
Next Hop	Displays the address of the BGP next hop.
Metric	Displays the BGP metric.
LocPrf	Displays the local preference.
Weight	Displays the BGP weight.
Path	Displays the BGP path per route.

The following example shows how to display a table of labels for NLRIs that have a route-distinguisher value of 100:1:

```
Router# show ip bgp vpnv4 rd 100:1 tags

    Network     Next Hop      In tag/Out tag
Route Distinguisher: 100:1 (vrf1)
   2.0.0.0          10.20.0.60      34/notag
   10.0.0.0         10.20.0.60      35/notag
   12.0.0.0         10.20.0.60      26/notag
                    10.20.0.60      26/notag
   13.0.0.0         10.15.0.15      notag/26
```

Table 2 describes the fields shown in the example.

Table 2 show ip bgp vpnv4 rd tags Field Descriptions

Field	Description
Network	Displays the network address from the BGP table.
Next Hop	Displays the BGP next hop address.
In Tag	Displays the label (if any) assigned by this router.
Out Tag	Displays the label assigned by the BGP next hop router.

The following example shows VPNv4 routing entries for the VRF called vrf1:

```
Router# show ip bgp vpnv4 vrf vrf1

BGP table version is 18, local router ID is 14.14.14.14
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
Route Distinguisher: 100:1 (vrf1)
*> 11.0.0.0         50.0.0.1 0 0 101 i
*>i12.0.0.0         13.13.13.13 0    100 0 102 i
*> 50.0.0.0         50.0.0.1 0 0 101 i
*>i51.0.0.0         13.13.13.13 0    100 0 102 i
```

Table 3 describes the fields shown in the example.

Table 3 show ip bgp vpnv4 Field Descriptions

Field	Description
Network	Displays network address from the BGP table.
Next Hop	Displays address of the BGP next hop.
Metric	Displays the BGP metric.
LocPrf	Displays the local preference.
Weight	Displays the BGP weight.
Path	Displays the BGP path per route.

Related Commands

Command	Description
show ip vrf	Displays VRFs and associated interfaces.

show ip cef vrf

To display the Cisco Express Forwarding (CEF) forwarding table associated with a Virtual Private Network (VPN) routing/forwarding instance (VRF), use the **show ip cef vrf** command in privileged EXEC mode.

show ip cef vrf *vrf-name* [*ip-prefix* [*mask* [**longer-prefixes**]] [**detail**] [*output-modifiers*]] [*interface interface-number*] [**adjacency** [*interface interface-number*] [**detail**] [**discard**] [**drop**] [**glean**] [**null**] [**punt**] [*output-modifiers*]] [**detail** [**output-modifiers**]] [**non-recursive** [**detail**] [*output-modifiers*]] [**summary** [*output-modifiers*]] [**traffic** [**prefix-length**] [*output-modifiers*]] [**unresolved** [**detail**] [*output-modifiers*]]

Syntax Description

vrf-name	Name assigned to the VRF.
ip-prefix	(Optional) IP prefix of entries to show, in dotted decimal format (A.B.C.D).
mask	(Optional) Mask of the IP prefix, in dotted decimal format.
longer-prefixes	(Optional) Displays table entries for all of the more specific routes.
detail	(Optional) Displays detailed information for each CEF table entry.
output-modifiers	(Optional) For a list of associated keywords and arguments, use context-sensitive help.
interface	(Optional) Type of network interface to use: ATM, Ethernet, Loopback, POS (packet over SONET) or Null.
interface-number	Number identifying the network interface to use.
adjacency	(Optional) Displays all prefixes resolving through adjacency.
discard	(Optional) Discards adjacency.
drop	(Optional) Drops adjacency.
glean	(Optional) Gleans adjacency.
null	(Optional) Null adjacency.
punt	(Optional) Punts adjacency.
non-recursive	(Optional) Displays only nonrecursive routes.
summary	(Optional) Displays a CEF table summary.
traffic	(Optional) Displays traffic statistics.
prefix-length	(Optional) Displays traffic statistics by prefix size.
unresolved	(Optional) Displays only unresolved routes.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.

Release	Modification
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Used with only the *vrf-name* argument, the **show ip cef vrf** command shows a shortened display of the CEF table.

Used with the **detail** keyword, the **show ip cef vrf** command shows detailed information for all CEF table entries.

Examples

This example shows the forwarding table associated with the VRF called vrf1:

```
Router# show ip cef vrf vrf1

Prefix              Next Hop            Interface
0.0.0.0/32          receive
11.0.0.0/8          50.0.0.1            Ethernet1/3
12.0.0.0/8          52.0.0.2            POS6/0
50.0.0.0/8          attached            Ethernet1/3
50.0.0.0/32         receive
50.0.0.1/32         50.0.0.1            Ethernet1/3
50.0.0.2/32         receive
50.255.255.255/32   receive
51.0.0.0/8          52.0.0.2            POS6/0
224.0.0.0/24        receive
255.255.255.255/32  receive
```

Table 4 describes the fields shown in the example.

Table 4 show ip cef vrf Field Descriptions

Field	Description
Prefix	Specifies the network prefix.
Next Hop	Specifies the BGP next hop address.
Interface	Specifies the VRF interface.

Related Commands

Command	Description
show ip route vrf	Displays the IP routing table associated with a VRF.
show ip vrf	Displays VRF interfaces.

show ip protocols vrf

To display the routing protocol information associated with a Virtual Private Network (VPN) routing/forwarding instance (VRF), use the **show ip protocols vrf** command in privileged EXEC mode.

show ip protocols vrf *vrf-name*

Syntax Description

vrf-name	Name assigned to a VRF.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC mode

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use this command to display routing information associated with a VRF.

Examples

The following example displays information about a VRF called vpn2:

```
Router# show ip protocols vrf vpn2

Routing Protocol is "bgp 100"
  Sending updates every 60 seconds, next due in 0 sec
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Redistributing:connected, static
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
    13.13.13.13          200      02:20:54
    18.18.18.18          200      03:26:15
  Distance:external 20 internal 200 local 200
```

Table 5 describes the fields shown in the example.

Table 5 show ip protocols vrf Field Descriptions

Field	Description
Gateway	Displays the IP address of the router identifier for all routers in the network.
Distance	Displays the metric used to access the destination route.
Last Update	Displays the last time the routing table was updated from the source.

Related Commands

Command	Description
show ip vrf	Displays VRF interfaces.

show ip route vrf

To display the IP routing table associated with a Virtual Private Network (VPN) routing/forwarding instance (VRF), use the **show ip route vrf** command in privileged EXEC mode.

show ip route vrf *vrf-name* [**connected**] [*protocol* [*as-number*] [*tag*] [*output-modifiers*]]
[**list** *number* [*output-modifiers*]] [**profile**] [**static** [*output-modifiers*]]
[**summary** [*output-modifiers*]] [**supernets-only** [*output-modifiers*]]
[**traffic-engineering** [*output-modifiers*]]

Syntax Description

vrf-name	Name assigned to the VRF.
connected	(Optional) Displays all connected routes in a VRF.
protocol	(Optional) To specify a routing protocol, use one of the following keywords: **bgp**, **egp**, **eigrp**, **hello**, **igrp**, **isis**, **ospf**, or **rip**.
as-number	(Optional) Autonomous system number.
tag	(Optional) Cisco IOS routing area label.
output-modifiers	(Optional) For a list of associated keywords and arguments, use context-sensitive help.
list *number*	(Optional) Specifies the IP access list to display.
profile	(Optional) Displays the IP routing table profile.
static	(Optional) Displays static routes.
summary	(Optional) Displays a summary of routes.
supernets-only	(Optional) Displays supernet entries only.
traffic-engineering	(Optional) Displays only traffic-engineered routes.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

This command displays specified information from the IP routing table of a VRF.

Examples

This example shows the IP routing table associated with the VRF called vrf1:

```
Router# show ip route vrf vrf1

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

B    51.0.0.0/8 [200/0] via 13.13.13.13, 00:24:19
C    50.0.0.0/8 is directly connected, Ethernet1/3
B    11.0.0.0/8 [20/0] via 50.0.0.1, 02:10:22
B    12.0.0.0/8 [200/0] via 13.13.13.13, 00:24:20
```

This example shows BGP entries in the IP routing table associated with the VRF called vrf1:

```
Router# show ip route vrf vrf1 bgp

B  51.0.0.0/8 [200/0] via 13.13.13.13, 03:44:14
B  11.0.0.0/8 [20/0] via 51.0.0.1, 03:44:12
B  12.0.0.0/8 [200/0] via 13.13.13.13, 03:43:14
```

Related Commands

Command	Description
show ip cef vrf	Displays the CEF forwarding table associated with a VRF.
show ip vrf	Displays VRFs and associated interfaces.

show ip vrf

To display the set of defined Virtual Private Network (VPN) routing/forwarding instances (VRFs) and associated interfaces, use the **show ip vrf** command in privileged EXEC mode.

show ip vrf [{**brief** | **detail** | **interfaces**}] [*vrf-name*] [*output-modifiers*]

Syntax Description

brief	(Optional) Displays concise information on the VRF(s) and associated interfaces.
detail	(Optional) Displays detailed information on the VRF(s) and associated interfaces.
interfaces	(Optional) Displays detailed information about all interfaces bound to a particular VRF, or any VRF.
vrf-name	Name assigned to a VRF.
output-modifiers	(Optional) For a list of associated keywords and arguments, use context-sensitive help.

Defaults

When no optional parameters are specified, the command shows concise information about all configured VRFs.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use this command to display information about VRFs. Two levels of detail are available: use the **brief** keyword or no keyword to display concise information, or use the **detail** keyword to display all information. To display information about all interfaces bound to a particular VRF, or to any VRF, use the **interfaces** keyword.

Examples

This example shows brief information for the VRFs currently configured:

```
Router# show ip vrf

  Name                    Default RD          Interfaces
  vrf1                    100:1               Ethernet1/3
  vrf2                    100:2               Ethernet0/3
```

Table 6 describes the fields shown in the example.

Table 6 show ip vrf Field Descriptions

Field	Description
Name	Specifies the VRF name.
Default RD	Specifies the default route distinguisher.
Interfaces	Specifies the network interfaces.

This example shows detailed information for the VRF called vrf1:

```
Router# show ip vrf detail vrf1

VRF vrf1; default RD 100:1
 Interfaces:
  Ethernet1/3
 Connected addresses are in global routing table
 Export VPN route-target communities
  RT:100:1
 Import VPN route-target communities
  RT:100:1
 No import route-map
```

Table 7 describes the fields shown in this example.

Table 7 show ip vrf detail Field Descriptions

Field	Description
Interfaces	Specifies the network interfaces.
Export	Specifies VPN route-target export communities.
Import	Specifies VPN route-target import communities.

This example shows the interfaces bound to a particular VRF:

```
router# show ip vrf interfaces

Interface        IP-Address       VRF                        Protocol
Ethernet2        130.22.0.33      blue_vrf                   up
Ethernet4        130.77.0.33      hub                        up
router#
```

Table 8 describes the fields shown in the example.

Table 8 show ip vrf interfaces Field Descriptions

Field	Description
Interface	Specifies the network interfaces for a VRF.
IP-Address	Specifies the IP address of a VRF interface.
VRF	Specifies the VRF name.
Protocol	Displays the state of the protocol (up/down) for each VRF interface.

Related Commands

Command	Description
import map	Configures an import route map for a VRF.
ip vrf	Enters VRF configuration mode.
ip vrf forwarding	Associates a VRF with an interface or subinterface.
rd	Configures a default RD for a VRF.
route-target	Configures import and export extended community attributes for the VRF.

show mpls forwarding vrf

To display label forwarding information for advertised Virtual Private Network (VPN) routing/forwarding instance (VRF) routes, use the **show mpls forwarding vrf** command in privileged EXEC mode. To disable the display of label forwarding information, use the **no** form of this command.

show mpls forwarding vrf *vrf-name* [*ip-prefix/length* [*mask*]] [**detail**] [*output-modifiers*]

no show mpls forwarding vrf *vrf-name* [*ip-prefix/length* [*mask*]] [**detail**] [*output-modifiers*]

Syntax Description

vrf-name	Displays NLRIs associated with the named VRF.
ip-prefix/length	(Optional) IP prefix address (in dotted decimal format) and length of mask (0 to 32).
mask	(Optional) Destination network mask, in dotted decimal format.
detail	(Optional) Displays detailed information on the VRF routes.
output-modifiers	(Optional) For a list of associated keywords and arguments, use context-sensitive help.

Defaults

No default behavior or values.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(21)ST	This command was modified to reflect new MPLS IETF terminology and CLI command syntax and was integrated into Cisco IOS 12.0(21)ST.
12.0(22)S	This command was integrated into Cisco IOS 12.0(22)S.
12.0(23)S	This command was integrated into Cisco IOS 12.0(23)S.
12.2(13)T	This command was integrated into Cisco IOS 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS 12.2(14)S.

Usage Guidelines

Use this command to display label forwarding entries associated with a particular VRF or IP prefix.

Examples

The following example shows label forwarding entries that correspond to the VRF called vpn1:

```
Router# show mpls forwarding vrf vpn1 detail

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
35     24          32.0.0.0/8[V]     0          Et0/0/4    42.0.0.1
        MAC/Encaps=14/22, MRU=1496, Tag Stack{24 19}
        00D006FEDBE100D0974988048847 0001800000013000
        VPN route: vpn1
        No output feature configured
    Per-packet load-sharing
```

Related Commands

Command	Description
show ip cef vrf	Displays VRFs and associated interfaces.
show mpls forwarding-table	Displays the contents of the LFIB.

Glossary

BGP—Border Gateway Protocol. Interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined in RFC 1163.

CEF—Cisco Express Forwarding. An advanced Layer 3 IP switching technology. CEF optimizes network performance and scalability for networks with large and dynamic traffic patterns.

CE router—customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router. CE routers are not aware of associated VPNs.

CoS—class of service. A feature that provides scalable, differentiated types of service across an MPLS network.

GRE—generic routing encapsulation. A tunneling protocol developed by Cisco that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork. By connecting multiprotocol subnetworks in a single-protocol backbone environment, IP tunneling that uses GRE allows network expansion across a single-protocol backbone environment.

IGP—Interior Gateway Protocol. An Internet protocol used to exchange routing information within an autonomous system. Examples of common IBGPs include IGRP, OSPF, and RIP.

IS-IS—Intermediate System-to-Intermediate System. OSI link-state hierarchical routing protocol in which ISs (routers) exchange routing information based on a single metric to determine network topology.

LFIB—label forwarding information base. A data structure and way of managing forwarding in which destinations and incoming labels are associated with outgoing interfaces and labels.

LSP—label-switched path. A sequence of hops (R0...Rn) in which a packet travels from R0 to Rn through label switching mechanisms. A label-switched path can be established dynamically, based on normal routing mechanisms, or through configuration.

LSP tunnel—label-switched path tunnel. A configured connection between two routers, in which MPLS is used to carry the packet.

MPLS—Multiprotocol Label Switching. An emerging industry standard. MPLS is a switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

NLRI—Network Layer Reachability Information. BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address, community values, and other information.

PE router—provider edge router. A router that is part of a service provider's network connected to a customer edge (CE) router. All VPN processing occurs in the PE router.

RD—route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

RIP—Routing Information Protocol. An IGP used to exchange routing information within an autonomous system, RIP uses hop count as a routing metric.

traffic engineering—The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

traffic engineering tunnel—A label-switched path tunnel that is used for engineering traffic. It is set up through means other than normal Layer 3 routing and is used to direct traffic over a path different from the one that Layer 3 routing would cause it to take.

tunneling—Architecture providing the services necessary to implement any standard point-to-point data encapsulation scheme.

VPN—Virtual Private Network. A secure IP-based network that shares resources on one or more physical networks. A VPN contains geographically dispersed sites that can communicate securely over a shared backbone.

VPNv4—Indicates a VPN-IPv4 prefix. These prefixes are customer VPN addresses, each of which has been made unique by the addition of an 8-byte route distinguisher.

VRF—VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.